49 EXPLOSIVE GAMES for the ZX81

Edited by
Tim Hartnell

49 EXPLOSIVE GAMES for the ZX81

Edited by
Tim Hartnell

Published in Great Britain by:

INTERFACE,

44 – 46 Earls Court Road,
London W8 6EJ.

ISBN 0 907563 08 2

Distributed in Scandinavia by:
ZX DATA
Box 44
Eu Dk 2650 Hvidovre
DENMARK

First printing – December 1981

This book is dedicated to Colin Hughes

OTHER PUBLICATIONS:

GETTING ACQUAINTED WITH YOUR ACORN ATOM
GETTING ACQUAINTED WITH YOUR ZX81
MASTERING MACHINE CODE ON YOUR ZX81
GETTING ACQUAINTED WITH YOUR VIC 20
THE GATEWAY GUIDE TO THE ZX81 & ZX80

Printed in England by Commercial Colour Press, London E7 OHX.

Foreword — by Tim Hartnell

Explosive games. Sounds exciting, doesn't it? In this book you'll find many great, great games to keep you locked to your ZX81 for hours.

This book has listings for every game we thought you might want, including GALACTIC INTRUDERS, BREAKOUT, DRAUGHTS/CHECKERS, STAR TREK, DEATH MAZE, 4-IN-A-ROW and an 8K ADVENTURE-type program SMUGGLERS BOLD. As well, there are a host of new games, and adaptations of old favourites. Many of the programs will run in just 1K (including a simplified SPACE INVADERS-type program).

Some of the games are based on chance — the dreaded Sinclair random number generator — and others depend on skill, both yours and the computer. But we've tried to ensure that each and every program contains at least one programming technique which you'll be able to adapt for your own programs.

You can, if you like, just enter the programs as listed, and play them. However, you're likely to get much more enjoyment from working with them, altering them as you choose, deleting some sections, improving others, and so on, until the game carries your personal stamp. Many of the 1K games can be improved if you have extra memory. At the very least, the player prompts can be made more "user-friendly", and the rules explained more exactly.

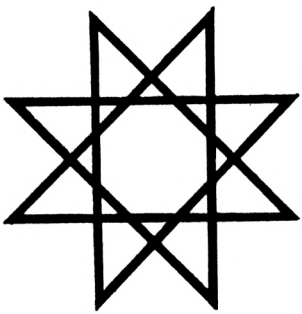
You should learn a lot from careful examinations of the listings. We've included a brief preface to each program, which is designed to explain —if it's not immediately evident from pressing RUN — how to play the game. For

some programs, we've also added a few words to clarify the algorithm used.

Colin Hughes of Luton and I wrote most of the games, with others from G D Charlton of Romford.

I'd like to thank these programmers, especially Colin, for their contributions, and also thank members of the National ZX80 and ZX81 Users' Club for the enthusiasm and friendliness they've shown in the past 18 months or so.

If we haven't yet had the pleasure of meeting you at a computer show, please come to the club stand at the next show and make yourself known. And if you haven't yet joined the club — which is probably the largest single users'

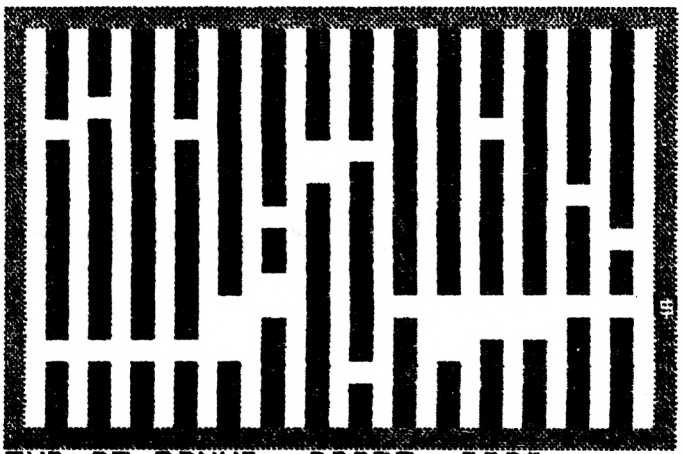


club in the UK — why not think about joining? We publish a monthly magazine INTERFACE, which is chock-a-block with ZX80, ZX81, ATOM and PROTON news, along with many programs for each machine, software, hardware and book reviews, contact addresses, letters from members, competitions and more. An application form for the club is near the back of this book.

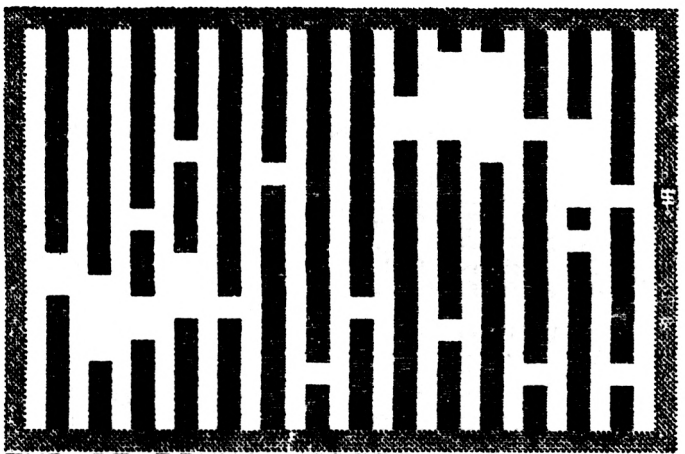
Looking forward to hearing from you,
TIM HARTNELL
London, November 1981.

DEATH MAZE

The ZX81 generates a maze, as you can see in the printout. You (the $ sign) start at the left hand side. You must get to the right as quickly as possible, using "Q" (up), "Z" (down) and "L" (right). The score is being decremented all the time.



```
END OF ROUND. SCORE:  6282
   BEST SO FAR:  6282
```



```
END OF ROUND. SCORE:  -852
   BEST SO FAR:  6516
```

If you hit a black line (rather than going through the space), the score drops dramatically. If you hit any border (except the right hand one), that round of the game is automatically terminated.

There is a highest score feature, and you should be able to get a score greater than 13504 with practice. Delete the FAST (10) and SLOW (9490) lines if you want to see the ZX81 build the DEATH MAZE between each round.

```
  3 LET U=0
 10 FAST
 15 FOR B=2 TO 28 STEP 2
 20 FOR A=0 TO 19
 30 PRINT AT A,B;"■"
 40 NEXT A
 45 PRINT AT RND*14+3,B;" "
 50 NEXT B
 60 FOR A=0 TO 30
 70 PRINT AT 0,A;"■";AT 19,A;"■
 80 NEXT A
 90 FOR B=1 TO 18
100 PRINT AT B,0;"■";AT B,30;"■
110 NEXT B
120 GOSUB 9000
125 GOTO 167
150 PRINT AT A,B;"■"
155 LET Z=Z-673
160 PRINT AT 20,0;"SCORE:  ";Z;"
    "
165 RETURN
167 LET Z$=INKEY$
170 IF Z$="" THEN LET Z$=A$
180 LET Z=Z-50
200 LET Y=A
210 LET X=B
215 REM  DELETE NEXT LINE FOR
         FASTER GAME
217 LET R=RND*RND*RND
220 LET A=A+(Z$="Z")-(Z$="Q")
230 LET B=B+(Z$="L")
232 PRINT AT Y,X;" "
233 IF PEEK (PEEK 16396+256*PEE
K 16397+33*A+B+1)=128 THEN GOSUB
 150
235 PRINT AT A,B;"$"
255 IF A>18 OR A<2 OR B<1 THEN
LET Z=INT (Z/3)
260 IF A>18 OR A<2 OR B<1 OR B>
29 THEN GOTO 510
```

```
 490 LET A$=Z$
 500 GOTO 167
 510 PRINT AT 20,0;"END OF ROUND
  SCORE:  ";Z
 520 IF Z>U THEN LET U=Z
 530 FOR G=1 TO 6
 540 PRINT AT 21,3;"BEST SO FAR:
 ";U
 545 PRINT AT A,B;"■";AT A,B;"$"
;AT A,B;"■"
 550 PRINT AT 21,3;"■■■■■"
 560 NEXT G
 570 CLS
 580 GOTO 10
9000 LET A=10
9010 LET B=1
9020 LET Z=20000
9030 LET Y=A
9040 LET X=B
9050 LET A$="Z"
9490 SLOW
9500 RETURN
```

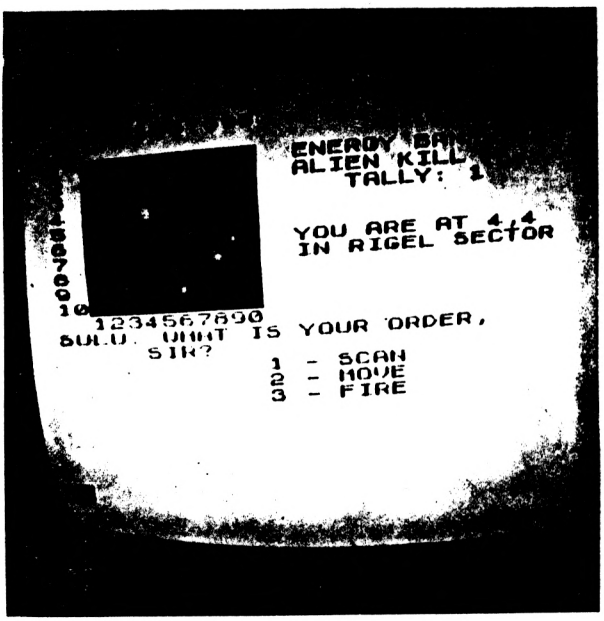
ASTER-DIVE

You are in control of a swinging line of asterisks, which you must guide -- using "M" and "Z" -- through randomly appearing blobs. The longer you last, the higher your score, which appears at the end of the game. You should be able to get more than 275. This is for a 1K ZX81.

```
  10 LET X=PI+PI
  15 LET U=X/X
  20 LET Y=X+PI
  30 LET Z=Y/Y
  40 PRINT AT Y,X;"*"
  50 SCROLL
  60 LET X=X+(INKEY$="M" AND X<3
0)-(INKEY$="Z" AND X>PI)
  62 LET U=U+Z
  65 IF RND>.7 THEN GOTO 40
  70 PRINT AT RND*15+5,RND*30;"■
"
  80 PRINT AT RND*15+5,RND*30;"■
"
  90 PRINT AT Y,X;
 110 IF PEEK (PEEK 16398+256*PEE
K 16399)<>8 THEN GOTO 40
 120 PRINT U
```

STAR TREK

This is a fascinating, and somewhat addictive game, in which you patrol a sector of the galaxy, in charge of a space ship crewed by such well-known spacers as Dr Spock, Scottie, Lt. Uhura, Chekov and Sulu. (Note that these names, and the name STAR TREK, are trademarks of Paramount Pictures Corporation.) There are 20 or so aliens in this sector of the galaxy. You are the dollar sign on the display.



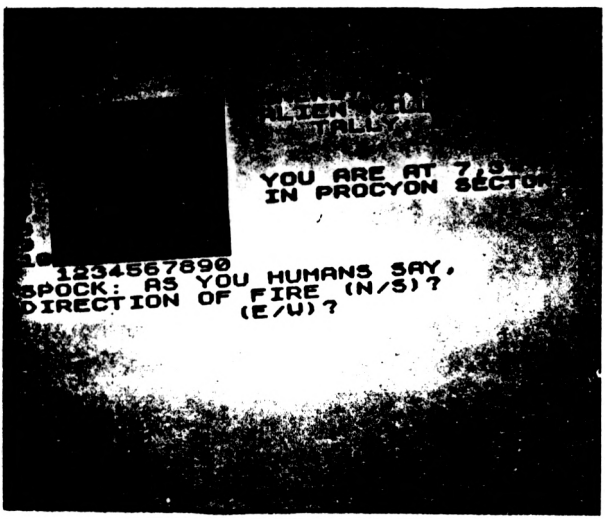
At any time, you move, scan or fire. You can move one square at a time north/south or east/west, and can fire only into the next square.

Your scanners operate in two ways:
SHORT RANGE looks into the eight squares immediately surrounding you; and
LONG RANGE looks, in the specified direction, at a single

square two squares away from you.

If you land on an alien ship the game is over. When you hit an alien, an inverse asterisk appears in its location, and your ALIEN KILL TOTAL is incremented. If there is no alien in that square, an inverse X appears so you know not to bother with that square again. The aliens do not move around during the course of a game. You have limited reserves in your energy bank and must try and kill as many aliens as you can before your energy runs out.

Careful use of your scanners can make sure you (a) waste as few shots as possible; and (b) don't land on an alien. Note



that long-range scanning uses up more energy than does short-range.

An alien can only shoot back after you have fired your laser at it (which reveals your position) and if the alien is within a single square of you. Damage to your ship from an accurate alien shot is shown in energy terms (that is, energy is drained from your bank). The game continues until you land on an alien ship, or run out of energy.

```
  20 GOSUB 9000
  25 GOSUB 8000
  30 GOSUB 6950
  40 GOSUB 7000
  50 GOSUB 6950
  80 GOSUB 7500
 100 PRINT "WHAT IS YOUR ORDER,"
 120 PRINT TAB 5;"SIR?",,TAB 12;
"1 - SCAN";TAB 12;"2 - MOVE";TAB
 12;"3 - FIRE"
 140 INPUT D
 150 IF D<1 OR D>3 THEN GOTO 140
 155 GOSUB 6950
 160 GOSUB 1000*D
 170 FOR W=1 TO 30
 180 PRINT AT 20,5;"▮▮▮▮▮";AT 20
 5;"     "
 190 NEXT W
 500 GOTO 30
1020 PRINT TAB 4;"▮▮▮ SCANNER ▮▮▮"
1030 GOSUB 7500
1040 PRINT "CLOSE (1) OR ","LONG
-RANGE (2), SIR?"
1060 INPUT K
1080 LET E=E-10*K
1090 GOSUB 6950
1100 IF K=2 THEN GOTO 1500
1120 IF A(B+1,C)=1 OR A(B+1,C+1)
=1 OR A(B,C+1)=1 OR A(B-1,C)=1 O
R A(B-1,C-1)=1 OR A(B,C-1)=1 OR
A(B+1,C-1)=1 OR A(B-1,C+1)=1 THE
N PRINT Z$;" IN VICINITY,","SIR"
1140 RETURN
1500 GOSUB 7500
1520 PRINT AT 15,0;"DIRECTION: N
-1, S-2, E-3, W-4?"
1525 PRINT TAB 8;"(ENTER A NUMBE
R)"
1530 INPUT N
1540 LET Z=0
1560 IF N=1 AND A(B-2,C)=1 THEN
LET Z=1
1580 IF N=2 AND A(B+2,C)=1 THEN
LET Z=1
1600 IF N=3 AND A(B,C+2)=1 THEN
LET Z=1
1620 IF N=488 AND A(B,C-2)=1 THE
N LET Z=1
1630 GOSUB 7500
1640 PRINT "LONG-RANGE SCANNER R
EPORT IS"
1660 IF Z=1 THEN PRINT "POSITIVE
"
1680 IF Z=0 THEN PRINT "NEGATIVE
"
1700 RETURN
2020 LET E=E-50
```

```
2040 LET A(B,C)=0
2050 LET B(B,C)=0
2060 PRINT "DIRECTION (N/S)?"
2080 INPUT A$
2100 LET B=B-1
2120 IF A$="S" THEN LET B=B+2
2130 GOSUB 7500
2140 PRINT "NOW AT ";B;",";C
2160 PRINT TAB 12;"(E/W)?"
2180 INPUT A$
2200 LET C=C-1
2220 IF A$="E" THEN LET C=C+2
2240 PRINT "NOW AT ";B;",";C
2260 IF A(B,C)=1 THEN GOTO 5500
2270 LET A(B,C)=2
2280 LET B(B,C)=2
2290 GOSUB 8000
2300 RETURN
3000 REM FIRE
3010 GOSUB 7500
3020 PRINT "DIRECTION OF FIRE (N
/S)?"
3040 INPUT A$
3060 LET G=B-1
3080 IF A$="N" THEN LET G=G+2
3100 PRINT TAB 12;"(E/W)?"
3120 LET F=C-1
3140 INPUT A$
3160 IF A$="E" THEN LET F=F+2
3180 LET E=E-100
3190 IF A(G,F)<>1 THEN GOTO 3300
3195 GOSUB 7500
3200 PRINT "YOU HIT THE ";Z$
3220 LET AL=AL+1
3260 LET B(G,F)=4
3290 RETURN
3300 GOSUB 7500
3305 LET B(G,F)=3
3310 PRINT "YOU MISSED, SIR"
3315 FOR G=1 TO 30
3317 NEXT G
3320 PRINT "THE ";Z$;" ARE","SHO
OTING BACK"
3330 FOR G=1 TO 30
3335 NEXT G
3337 GOSUB 6950
3340 IF RND>.6 THEN GOTO 3400
3360 PRINT "THEY HIT US, SIR"
3370 GOSUB 8000
3380 LET E=E-100*RND
3390 RETURN
3420 PRINT "THE ";Z$;" MISSED, S
IR"
3460 RETURN
3800 GOSUB 6950
3805 SCROLL
3810 PRINT "ENERGY BANKS EXHAUST
ED"
```

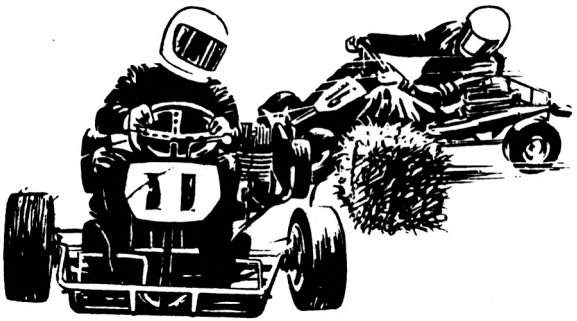
```
3815 SCROLL
3820 PRINT "YOU KILLED ";AL;" AL
IEN";
3830 IF AL<>1 THEN PRINT "S"
3850 SCROLL
3860 PRINT "ON THIS MISSION"
3870 SCROLL
3880 PRINT "YOUR COMMANDER RATIN
G IS ";INT (AL/8*100)
3890 GOTO 3805
5000 REM END
5500 GOSUB 6950
5520 PRINT AT 15,0;"YOUR SHIP HA
S LANDED ON A",Z$;" VESSEL"
5540 PRINT AT 15,0;"[illegible]
[illegible]"
5560 GOTO 5520
5900 STOP
6950 PRINT AT 13,0;"
```

```
6955 PRINT AT 13,0;
6970 RETURN
7000 REM STATUS
7020 PRINT AT 2,14;"ENERGY BANK:
 ";INT E;" "
7030 IF E<1 THEN GOTO 3800
7040 IF AL>0 THEN PRINT AT 3,14;
"ALIEN KILL";AT 4,17;"TALLY: ";A
L
7060 PRINT AT 7,14;"YOU ARE AT "
;B;",";C
7070 PRINT AT 8,14;"
"
7075 PRINT AT 8,14;"IN ";
7080 GOSUB 8500
7100 PRINT " SECTOR"
7120 PRINT AT 12,0;
7490 RETURN
7500 LET R=INT (RND*5)
7520 IF R=0 THEN PRINT "SPOCK: A
S YOU HUMANS SAY,"
7540 IF R=1 THEN PRINT "SCOTT: "
;
7560 IF R=2 THEN PRINT "LT. UHUR
A: ";
7580 IF R=3 THEN PRINT "CHEKOV:
";
7600 IF R=4 THEN PRINT "SULU: ";
7900 RETURN
7999 STOP
8000 REM PRINT OUT
8005 PRINT AT 0,0;
```

```
8010 PRINT "  1234567890"
8020 FOR Q=1 TO 10
8025 PRINT AT Q,13;"▒";AT Q,13;"
 "
8030 IF Q<10 THEN PRINT Q;" ";
8035 IF Q=10 THEN PRINT Q;
8040 FOR P=1 TO 10
8060 IF B(Q,P)=0 THEN PRINT "▒";
8080 IF B(Q,P)=2 THEN PRINT "#";
8100 IF B(Q,P)=3 THEN PRINT "*";
8120 IF B(Q,P)=4 THEN PRINT "▒";
8160 NEXT P
8200 NEXT Q
8210 PRINT
8220 PRINT "  1234567890"
8490 RETURN
8500 REM SECTOR
8520 LET Q=8*C
8540 IF Q<10 THEN PRINT "ANTARES
";
8560 IF Q>9 AND Q<20 THEN PRINT
"RIGEL";
8580 IF Q>19 AND Q<30 THEN PRINT
 "PROCYON";
8600 IF Q>29 AND Q<40 THEN PRINT
 "VEGA";
8620 IF Q>39 AND Q<50 THEN PRINT
 "CANOPUS";
8640 IF Q>49 AND Q<60 THEN PRINT
 "ALTAIR";
8660 IF Q>59 AND Q<70 THEN PRINT
 "SAGITTARIUS";
8680 IF Q>69 AND Q<80 THEN PRINT
 "POLLUX";
8700 IF Q>79 AND Q<90 THEN PRINT
 "SIRIUS";
8720 IF Q>89 THEN PRINT "BETELGE
USE";
8740 RETURN
8999 STOP
9000 DIM A(10,10)
9020 DIM B(10,10)
9060 FOR A=1 TO 20
9080 LET X=INT (RND*10+1)
9100 LET Y=INT (RND*10+1)
9120 LET A(X,Y)=1
9140 NEXT A
9160 LET B=5
9180 LET C=5
9200 LET A(B,C)=2
9220 LET B(B,C)=2
9240 LET AL=0
9260 LET E=RND
9280 IF E<.33 THEN LET Z$="BRARK
ONS "
9300 IF E>.33 AND E<.66 THEN LET
 Z$="WRERKTONIONS "
9320 IF E>.66 THEN LET Z$="POLLU
XIANS "
9340 LET E=1000+2000*RND
9900 RETURN
```

DODGEM

This is more fun than most car-driving programs. You, an inverse Y, have a limited time in which to hit as many asterisks as you can. "Z" and "C" control your car.



```
    1 LET H=0
    5 LET P=5
   10 FOR I=1 TO 8
   20 PRINT TAB 15;" "
   30 NEXT I
   35 FOR I=1 TO 50
   40 LET A=INT (P+RND*5-2)
   41 IF A<0 OR A>14 THEN LET A=I
NT (RND*15)
   45 PRINT AT 8,A;"**";AT 8,15;"
";AT 3,P;" "
   50 SCROLL
   70 IF INKEY$="Z" AND P<>0 THEN
 LET P=P-1
   80 IF INKEY$="C" AND P<>14 THE
N LET P=P+1
  101 LET P1=PEEK 16396+256*PEEK
16397
  102 IF PEEK (P1+1+P+3*17)=23 TH
EN GOTO 400
  105 PRINT AT 3,P;""
  106 PAUSE 15
  110 NEXT I
  120 PRINT H
  130 STOP
  400 PRINT AT 2,0;"******BANG***
***"
  410 PRINT AT 3,P;""
  420 PAUSE 50
  430 LET H=H+1
  450 GOTO 40
```

GALACTIC INTRUDERS

The INTRUDERS in this game have a fearful weapon - a horrid black bird which swoops down at you without warning. You use "5" and "8" to move right and left, and "1" to fire. Your score increases with every INTRUDER you destroy, and the INTRUDER gets a score every time a black bird swoops onto you. The tally at the end is based on the difference between your score and the INTRUDERS' score. There is a highest score feature. This game needs more than 1K.

```
  50 LET U=0
 100 GOSUB 9000
 400 FOR N=1 TO 40
 410 IF N=1 THEN GOTO 1065
 500 LET M=0
 700 LET Z$=INKEY$
 800 IF Z$="8" THEN LET B=B+1
 900 IF Z$="5" THEN LET B=B-1
 950 IF RND>.4 THEN LET M=1
1000 IF Z$<>"1" THEN GOTO 1055
1010 FOR A=19 TO 5 STEP -2
1020 IF M=0 THEN GOTO 1056
1022 LET M=0
1025 LET Q=B
1027 IF A$(Q)=" " THEN GOTO 1056
1030 FOR E=3 TO 19 STEP 4
1040 PRINT AT E,Q;"▀▀";AT E,Q;"
  "
1045 IF INKEY$="5" THEN LET B=B-
1
1046 IF INKEY$="8" THEN LET B=B+
1
1047 PRINT AT 20,B-1;" ▄▀▄ "
1050 NEXT E
1052 IF B=Q THEN LET C=C+1
1053 IF B=Q THEN GOSUB 5000
1055 IF M=1 THEN GOTO 1022
1056 LET A$=A$(2 TO )+A$(1)
1057 IF Z$<>"1" THEN GOTO 1065
1060 PRINT AT A,B+1;"*";AT A,B+1
;" "
1065 PRINT AT 2,2;A$;AT 4,0;A$;A
T 6,1;A$;AT 20,B;" ▄▀▄ "
1067 IF Z$<>"1" THEN GOTO 2500
1070 NEXT A
1075 IF A$(B+1)<>" " THEN LET S=
S+1
1080 PRINT AT 4,B;"*";AT 0,0;"YO
U ";S*641
```

```
2000 LET A$(B+1)=" "
2500 NEXT N
2520 PRINT AT 0,16;"ME ";C*439
3000 PRINT AT 10,10;"END OF ROUN
D"
3005 LET S=INT (S-C/2)
3010 IF S>U THEN LET U=S
3020 PRINT "YOUR HIGHEST TALLY:
";U*641
3080 FOR N=1 TO 60
3090 NEXT N
3095 CLS
4000 GOTO 100
5000 FOR J=1 TO 3
5010 PRINT AT 10,10;"████";AT 10
,10;"BOOM";AT 10,10;"    "
5020 PRINT AT 20,Q+1;"█";AT 20,Q
+1;"█";AT 20,Q+1;"X";AT 20,Q+1;"
M";AT 20,Q+1;"█"
5050 NEXT J
5060 PRINT AT 0,0;"███ ";S*641,"
██ ";C*439
5100 RETURN
8990 STOP
9010 LET Z$=""
9020 LET S=0
9050 LET A$="█ █ █ █ █ █ ███
 █ █ █ "
9070 LET B=15
9090 LET C=0
9990 RETURN
```

18TH HOLE

In this 1K game you see a ball and a hole. You enter the strength of the shot you think will hole the ball (1 to 100). The ball will then move and, if you've been accurate, you will actually see it drop into the hole. You might like to add a "score card" feature if you have more memory.

```
  5 LET S=0
 10 LET J=INT (RND*15)
 20 GOSUB 200
 30 PRINT "██████████████████████
██████ █████"
 40 PRINT AT 9,7;"STROKE?"
 50 INPUT A
 52 PRINT AT 9,7;"       "
 55 PRINT AT 17,J;" "
 60 LET J=J+INT (A/(6+RND))
 70 GOSUB 200
```

```
  80 LET S=S+1
  85 PRINT AT 5,12;"COUNT ";S
  90 IF J<27 THEN GOTO 40
 100 IF J>27 THEN GOTO 150
 105 PRINT AT 17,27;" ";AT 18,27
;"O"
 110 PRINT "YOU SCORED ";S;" ON
THAT HOLE"
 120 GOTO 160
 150 PRINT "HOPELESS, YOU OVERSH
OT"
 160 FOR G=1 TO 100
 170 NEXT G
 180 CLS
 190 RUN
 200 PRINT AT 17,J;"O"
 210 RETURN
```

CHECKERS/DRAUGHTS

This checkers program, adapted for a 16K ZX81 by Tim Hartnell from a 1K ZX80 program written by G. D. Charlton, of Romford, plays fairly well at the beginning, but falls away towards the end of the game. You may like to "teach" it to play a little better. You move by entering the numbers which flank the square you're moving from (such as 53, in which the 5 is along the right hand edge of the board, and the 3 is along the top), then NEWLINE, then the square you're moving to. The computer will then decide on its move (in the FAST mode). The ZX81 goes back into SLOW to reprint the board, changing both moves as it does so.

After each move you'll be asked if you can move again. If not, just press NEWLINE and the ZX81 will move. If you can, press any key before pressing NEWLINE and you'll be able to move again. The ZX81 makes multiple jumps automatically.

```
  10 REM AAA■23456789■1■X■X■X■X1
2X■X■X■X■23■X■X■X■X34■■■■■■45■
■■■■■ 560■0■0■0■67■0■0■0■0780■0
■0■0■8■23456789■
  11 REM AAA■23456789■1■X■X■X■X1
2X■X■X■X■23■X■X■X■X34■■■■■■45■
■■■■■ 560■0■0■0■67■0■0■0■0780■0
■0■0■8■23456789■
  12 FAST
  17 GOSUB 5000
  19 GOTO 430
  20 LET R=PEEK B
  30 LET S=PEEK (B+D(X))
  40 LET T=PEEK (B+2*D(X))
  50 RETURN
  55 LET Z=0
  60 FOR B=16528 TO 16607
  70 FOR X=1 TO 4
  80 GOSUB 20
  90 IF ((X<3 AND R=61) OR R=13)
 AND (S=52 OR S=12) AND T=0 THEN
 GOTO 300
 100 NEXT X
 110 NEXT B
 120 FOR A=1 TO 400
 130 LET B=INT (RND*80)+16528
 140 FOR X=1 TO 4
 150 GOSUB 20
 160 IF ((X<3 AND R=61) OR R=13)
 AND S=0 THEN GOTO 400
 170 NEXT X
 180 NEXT A
 190 PRINT "YOU WIN";0__
 200 SLOW
 202 PRINT AT 0,0;"FROM?"
 205 INPUT G
 207 PRINT AT 0,4;" ";G;" TO?"
 210 INPUT H
 215 PRINT AT 0,0;"              "
 220 POKE 16516+H,PEEK (G+16516)
 230 IF H<20 THEN POKE H+16516,1
2
 240 POKE 16516+G,0
 250 IF ABS (H-G)=18 OR ABS (H-G
)=22 THEN POKE 16516+(H+G)/2,0
 260 PRINT AT 17,0;"CAN YOU MOVE
 AGAIN?"
 270 INPUT A$
 275 PRINT AT 17,0;"
     "
 280 IF A$<>"" THEN GOTO 430
 285 FAST
 290 GOTO 60
 300 POKE B+2*D(X),R
 310 POKE B,0
 320 POKE B+D(X),0
 330 LET B=B+2*D(X)
 340 IF B>16597 THEN GOTO 490
```

```
  350 FOR X=1 TO 4
  360 GOSUB 20
  370 IF ((X<3 AND R=61) OR R=13)
  AND (S=52 OR S=12) AND T=0 THEN
  GOTO 300
  380 NEXT X
  390 GOTO 430
  400 POKE B+D(X),R
  410 POKE B,0
  420 IF B+D(X)>16597 THEN POKE B
+D(X),13
  425 SLOW
  430 PRINT AT 6,8;
  440 FOR A=16517 TO 16616
  450 PRINT CHR$ (PEEK A);
  460 IF 10*INT ((A+4)/10)=A+4 TH
EN PRINT TAB 8;
  470 NEXT A
  472 PRINT
  475 IF Z=1 THEN GOTO 55
  480 GOTO 200
  490 POKE B,13
  500 GOTO 430
 5000 DIM D(4)
 5010 LET D(1)=11
 5020 LET D(2)=9
 5030 LET D(3)=-9
 5040 LET D(4)=-11
 5050 FOR J=16626 TO 16725
 5060 POKE J-109,PEEK J
 5070 NEXT J
 5080 LET Z=1
 5090 REM "DRAUGHTS"    ADAPTED
T HARTNELL FROM ZX80 PROGRAM
 BY G D CHARLTON, ROMFORD
 5100 RAND
 6000 RETURN
```

MAHOGANY

The computer thinks of a number between one and nine (the top number displayed). You have to try and anticipate the next number it will think of, by touching that number. Your number will be displayed underneath the computer's number. Under this number is the count of how many goes you have had so far. The smaller this number when the game stops -- that is, when you successfully anticipate the ZX81's number -- the better. This fits within 1K.

```
  5 LET E=9
 10 LET W=INT (RND*9)+1
 15 LET Q=W/W
 25 LET Z=CODE INKEY$-28
 27 IF Z=-28 THEN PRINT AT 12,E
;"■"
 30 IF Z>0 AND Z<10 THEN PRINT
AT 12,E;Z
 40 LET Q=Q+Q/Q
 45 PRINT AT E,E;W
 50 IF W=Z THEN PRINT AT 14,8;"
>>";Q;A
 60 PRINT AT 14,E;Q
 70 LET M=INT (RND*5)-INT (RND*
5)
 80 IF M+W>0 AND M+W<10 THEN LE
T W=W+M
 90 GOTO 25
```

BREAKOUT

You control the action of the slide at the bottom of the screen with the "5" and "8" keys, trying to keep the ball bouncing as long as possible. You have nine balls, and you'll get an increase in score every time you manage to wipe out one of the blobs near the top of the screen. The score is changed on the screen each time you lose a ball. There are five rows of blobs to knock down. A perfect score is 3618. At the end of the game, you'll get a percentage "perfection" rating.

```
 10 GOSUB 9000
 20 GOSUB 8000
 30 IF PEEK (33*Y+X+1+PEEK 1639
6+256*PEEK 16397)=R THEN GOSUB 7
000
 50 PRINT AT Y,X;A$
 60 PRINT AT Y,X;" "
510 PRINT AT 20,M-1;"  ▀▀  "
530 IF Y=1 OR (Y=19 AND ABS (M-
X)<3) THEN LET Q=-Q
540 IF X=2 OR X=30 THEN LET W=-
W
545 LET M=M+(INKEY$="8")-(INKEY
$="5")
550 IF Y=20 THEN GOSUB 8000
600 LET Y=Y+Q
610 LET X=X+W
6000 GOTO 30
```

```
7000 PRINT AT Y,X;"■";AT Y,X;"*"
;AT Y,X;"■"
7010 IF Y<>1 THEN LET Q=-Q
7020 LET S=S+67
7500 RETURN
7999 STOP
8000 PRINT AT 19,0;"
                    "
8002 PRINT AT 20,0;"
                    "
8005 LET A=A+1
8010 PRINT AT 0,7;S
8015 IF A=166 THEN GOTO 9500
8020 LET A$=CHR$ (A)
8030 LET Y=18
8040 LET X=INT (RND*26+4)
8050 LET M=2
8060 LET Q=-1
8070 LET W=1
8075 IF RND>.5 THEN LET W=-1
8100 RETURN
8999 STOP
9000 PRINT "SCORE: "
9050 FOR J=1 TO 54
9060 PRINT "■ ";
9070 NEXT J
9080 LET A=156
9090 LET S=0
9100 LET R=136
9110 RETURN
9500 PRINT AT 10,2;"YOUR RATING
IS ";INT (S*1000/3618)/10;" PER
CENT"
9510 PRINT TAB 4;"OF GRAND MASTE
R STATUS"
```

CONEY ISLAND

You have to shoot at the little coney flying across the top of the screen. "5" and "8" move you left and right, and "0" activates your anti-coney gun. You have just 10 shots, although you can alter line 1035 if you want more or less. You're rewarded with a satisfying display every time you get a coney. At the end of the game the score is printed over and over again at random positions. A score of five or more is very good.


```
  5 LET F=0
 10 LET A=20
 15 LET K=2
 20 LET B=A-A
 22 LET P=B
 25 LET M=0
 30 LET C=B
 40 LET D=25
 45 LET S=D
 50 PRINT AT A,P;" "
 55 PRINT AT A,B;"▀"
 57 PRINT AT C,S;" "
 60 PRINT AT C,D;"▄"
 65 IF INKEY$="0" THEN GOSUB 10
00
 67 LET P=B
 70 LET B=B-(INKEY$="8")+(INKEY
$="5")
 75 LET S=D
 80 LET D=D+INT (K*RND)
 90 IF D<1 OR D>30 THEN LET K=-
K
100 IF D<1 THEN LET D=2
110 IF D>30 THEN LET D=29
150 GOTO 50
1000 LET F=F+1
1004 PRINT AT 8,25;"SHOT ";F
1005 FOR G=18 TO 3 STEP -3
1010 PRINT AT G,B;"*"
1015 PRINT AT G,B;" "
1020 NEXT G
1025 PRINT AT 8,25;"          "
1030 IF ABS (B)-ABS (D)=0 THEN G
OSUB 2000
1035 IF F=10 THEN GOTO 3000
1040 RETURN
2000 LET M=M+1
2005 PRINT AT 10,10;"YOUR SCORE
IS ";M
2007 FOR T=1 TO 6
2015 PRINT AT C,D-1;"▐  ▌"
2017 PRINT AT C,D-1;"▀  ▀"
2020 PRINT AT C,D-1;"    "
2022 PRINT AT 1,D;"▀"
2025 PRINT AT C,D-1;"    "
2027 PRINT AT 1,D;" "
2030 NEXT T
2040 PRINT AT 10,10;"
    "
2050 RETURN
3000 PRINT AT 0,0;"END OF GAME"
3010 PRINT AT RND*20,RND*15;"YOU
 SCORED ";M
3020 GOTO 3010
```

DALI

This is a very simple "ETCH-A-SKETCH" program for the 1K ZX81. You use the "5", "8", "6" and "7" keys to control the movement of the flashing dot to draw pictures of your choice.

```
  10 LET X=15
  20 LET Y=INT (X/PI)
  30 IF INKEY$="" THEN GOTO 30
  70 LET Y=Y-(INKEY$="5" AND Y>2
)+(INKEY$="8" AND Y<60)
  80 LET X=X-(INKEY$="6" AND X>2
)+(INKEY$="7" AND X<40)
 110 UNPLOT Y,X
 120 PLOT Y,X
 130 GOTO 70
```

PEEK-A-BOO, POKE-A-BOO

The game begins with a ball in motion, and you have to control the ball (using the "6" and "7" keys) to hit as many black squares as you can. There is a highest score feature, but — as the game gets harder as it progresses — it becomes more and more difficult to achieve higher scores as the game continues. When you're tired of playing this game according to the rules, try to MISS as many black squares as possible.

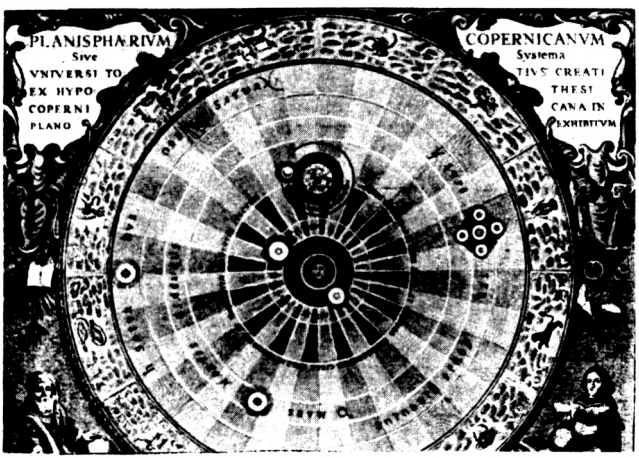
```
   5 LET U=0
  10 FOR Z=1 TO 60
  20 PRINT AT 3+RND*16,2+RND*29;
"■"
  30 NEXT Z
  40 LET E=16396
  50 LET F=16397
  60 LET T=0
  70 LET M=0
 100 LET X=10
 102 LET K=1
 105 LET B=X
 107 LET B$="0"
```

```
 110 LET Y=10
 111 LET A=X
 117 POKE 33*B+A+1+PEEK E+256*PE
EK F,0
 118 IF PEEK (33*Y+X+1+PEEK E+25
6*PEEK F)=128 THEN GOSUB 500
 120 POKE 33*Y+X+1+PEEK E+256*PE
EK F,52
 121 LET T=T+3
 122 LET A=X
 123 IF T>200 THEN GOSUB 500
 124 LET B=Y
 130 LET X=X+K
 135 IF X<2 OR X>30 THEN LET K=-
K
 136 LET A$=INKEY$
 137 IF A$="" THEN LET A$=B$
 138 LET Y=Y-(A$="7")+(A$="6")
 139 IF Y<2 THEN LET A$="6"
 140 IF Y>18 THEN LET A$="7"
 150 LET B$=A$
 170 GOTO 117
 500 LET M=M+1
 502 POKE 33*Y+X+1+PEEK E+256*PE
EK F,189
 505 PRINT AT 0,0;"YOUR SCORE IS
 ";M;AT 1,5;"TIME IS ";T
 513 FOR Z=1 TO 7
 515 POKE 33*Y+X+1+PEEK E+256*PE
EK F,189
 516 POKE 33*Y+X+1+PEEK E+256*PE
EK F,23
 518 POKE 33*Y+X+1+PEEK E+256*PE
EK F,52
 520 NEXT Z
 522 LET R=117*M
 525 PRINT AT 20,0;"TALLY IS ";R
 527 FOR H=1 TO 6
 528 POKE 33*Y+X+1+PEEK E+256*PE
EK F,189
 529 POKE 33*Y+X+1+PEEK E+256*PE
EK F,52
 530 NEXT H
 535 PRINT AT 20,0;"
 "
 540 IF T>200 THEN GOSUB 1000
 560 PRINT AT 0,0;"
                   "
 570 RETURN
1000 IF R>U THEN LET U=R
1010 PRINT AT 0,0;"GAME OVER - Y
OUR TALLY - ";R;AT 1,5;"BEST SO
FAR - ";U
1020 FOR H=1 TO 23
1022 POKE 33*Y+X+1+PEEK E+256*PE
EK F,189
```

```
1025 POKE 33*Y+X+1+PEEK E+256*PE
EK F,52
1030 NEXT H
1035 LET M=0
1040 LET T=0
1045 LET R=0
1050 RETURN
```

HUAMBO

There are two versions of this game listed. In the first one, the computer is the inverse letter "C" trying to get all its pieces from the top of the board (a 5 × 5 grid) to the bottom, before the human (the inverse H's) gets all his or her pieces to the top. In the second version, you are the inverse £ signs, trying to move from left to right and the computer is the inverse $ signs trying to move from the top of the screen to the bottom. There are no captures in this game, and both players can move one square in any



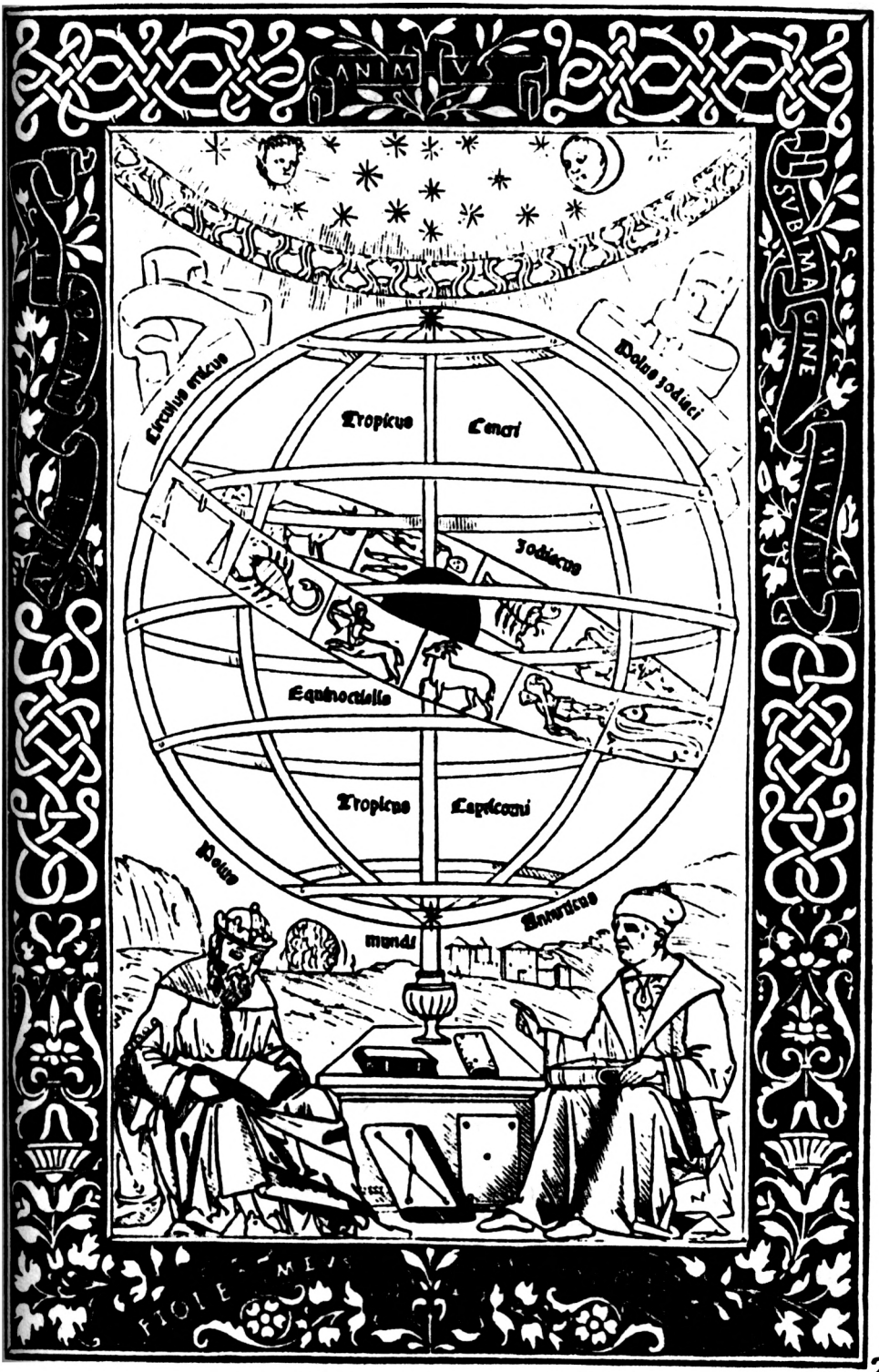
direction (forward, backward, up, down, or along the diagonals). You move by entering the square you're moving from and to in one string. For example, "E1D2" will move you from the E1 square to the D2 square. The best strategy is to try and block future moves by your opponent while at the same time getting as many of your men to the opposing side as you can.

```
    5 LET C=168
    7 LET H=173
   10 DIM A(25)
   20 DIM B(7)
   30 FOR A=3 TO 5
   40 LET A(A)=C
   50 NEXT A
   70 FOR A=21 TO 23
   80 LET A(A)=H
   90 NEXT A
  110 LET B(1)=5
  120 LET B(2)=4
  130 LET B(3)=6
  140 LET B(4)=-4
  150 LET B(5)=-5
  160 LET B(6)=-6
  165 GOSUB 1000
  170 FOR A=20 TO 1 STEP -1
  175 FOR J=1 TO 5
  180 LET B=INT (RND*3)+1
  185 IF 5*INT (A/5)=A AND B=3 TH
EN GOTO 210
  190 IF A(A)=C AND A(A+B(B))=0 T
HEN GOTO 270
  200 NEXT J
  210 NEXT A
  220 FOR A=7 TO 20
  230 FOR B=4 TO 6
  235 IF 5*(INT (A/5))=A AND B=4
THEN GOTO 250
  240 IF A(A)=C AND A(A+B(B))=0 T
HEN GOTO 270
  250 NEXT B
  260 NEXT A
  265 PRINT "I CONCEDE";W
  270 LET A(A+B(B))=C
  280 LET A(A)=0
  290 GOSUB 1000
  300 REM [redacted]
  310 INPUT A$
  320 LET D=5*(CODE A$-38)+CODE A
$(2)-28
  330 LET E=5*(CODE A$(3)-38)+COD
E A$(4)-28
  340 LET A(E)=H
  350 LET A(D)=0
  360 GOTO 165
1000 LET X=0
1010 LET Y=0
1012 PRINT AT 20,0;"[redacted]
[redacted]"
1013 LET Z=RND**RND
1015 PRINT AT 20,0;"-----------
---------"
1020 PRINT AT 5,0;"1 2 3 4 5"
1025 PRINT
1030 FOR A=1 TO 25
```

```
1040 PRINT CHR$ (A(A));"*";
1050 IF 5*INT (A/5)=A THEN PRINT
  " ";CHR$ (A/5+37);.....
1060 IF A(A)=C AND A>20 THEN LET
  X=X+1
1070 IF A(A)=H AND A<6 THEN LET
Y=Y+1
1080 NEXT A
1090 PRINT "1 2 3 4 5"
1095 PRINT
1100 PRINT "HUMAN ";Y;"    COMPUT
ER ";X
1101 IF Y>X THEN PRINT "YOU ARE
WINNING"
1102 IF X>Y THEN PRINT "----I AM
 WINNING"
1105 IF X=4 OR Y=4 THEN GOTO 112
0
1110 RETURN
1120 IF X>Y THEN PRINT "COMPUTER
";
1130 IF Y>X THEN PRINT "HUMAN";
1140 PRINT "WINS BY ";ABS (X-Y);
" POINTS"
```

```
   5 LET C=141
   7 LET H=140
  10 DIM A(25)
  20 DIM B(7)
  30 FOR A=3 TO 5
  40 LET A(A)=C
  50 NEXT A
  70 LET A(11)=H
  80 LET A(16)=H
  90 LET A(21)=H
 110 LET B(1)=5
 120 LET B(2)=4
 130 LET B(3)=6
 140 LET B(4)=-4
 150 LET B(5)=-5
 160 LET B(6)=-6
 165 GOSUB 1000
 170 FOR A=20 TO 1 STEP -1
 175 FOR J=1 TO 5
 180 LET B=INT (RND*3)+1
 185 IF 5*INT (A/5)=A AND B=3 TH
EN GOTO 210
 190 IF A(A)=C AND A(A+B(B))=0 T
HEN GOTO 270
 200 NEXT J
 210 NEXT A
```

```
 220 FOR A=7 TO 20
 230 FOR B=4 TO 6
 232 IF (A=6 OR A=11 OR A=16) AN
D B=4 THEN GOTO 250
 235 IF 5*(INT (A/5))=A AND B=4
THEN GOTO 250
 240 IF A(A)=C AND A(A+B(B))=0 T
HEN GOTO 270
 250 NEXT B
 260 NEXT A
 265 PRINT "I CONCEDE";W
 270 LET A(A+B(B))=C
 280 LET A(A)=0
 290 GOSUB 1000
 300 REM [redacted]
 310 INPUT A$
 320 LET D=5*(CODE A$-38)+CODE A
$(2)-28
 330 LET E=5*(CODE A$(3)-38)+COD
E A$(4)-28
 340 LET A(E)=H
 350 LET A(D)=0
 360 GOTO 165
1000 LET X=0
1010 LET Y=0
1012 PRINT AT 20,0;"[redacted]
[redacted]"
1013 LET Z=RND**RND
1015 PRINT AT 20,0;"------------
---------"
1020 PRINT AT 5,0;"1 2 3 4 5"
1025 PRINT
1030 FOR A=1 TO 25
1040 IF A(A)=0 THEN PRINT "[redacted] ";
1045 IF A(A)<>0 THEN PRINT CHR$
(A(A));" ";
1050 IF 5*INT (A/5)=A THEN PRINT
" ";CHR$ (A/5+37);;;
1060 IF A(A)=C AND A>20 THEN LET
 X=X+1
1070 IF A(A)=H AND 5*INT (A/5)=A
 THEN LET Y=Y+1
1080 NEXT A
1090 PRINT "1 2 3 4 5"
1095 PRINT
1100 PRINT "HUMAN ";Y;"   COMPUT
ER ";X
1101 IF Y>X THEN PRINT "YOU ARE
WINNING"
1102 IF X>Y THEN PRINT "----I AM
 WINNING"
1105 IF X=3 OR Y=3 THEN GOTO 112
0
1110 RETURN
1120 IF X>Y THEN PRINT "COMPUTER
";
1130 IF Y>X THEN PRINT "HUMAN";
1140 PRINT "WINS BY ";ABS (X-Y);
" POINTS"
```


ANIMVS
SVBLIMA GINE
Circulus arcticus
Polus zodiaci
Tropicus Cancri
Zodiacus
Equinoctialis
Tropicus Capricorni
Polus
mundi
Antarticus

HAPPY CHAPPY

HAPPY CHAPPY is a large face which bounces across the screen, an idiotic smile on his face. Hit any key as he passes over the marker in the middle of the screen, and the HAPPY CHAPPY stops, his smile changed to a scowl. You get to take 10 pot shots at 10 HAPPY CHAPPIES. This 1K game fits new ROM ZX80's and ZX81's. Run it in FAST.

```
 10 DIM A$(5,5)
 20 LET A$(1)="     "
 30 LET A$(2)="     "
 40 LET A$(3)="     "
 50 LET A$(4)="     "
 60 LET A$(5)="     "
 70 LET S=0
 80 FOR F=1 TO 10
 90 PRINT AT 5,13;"  "
100 PRINT TAB 13;"  "
110 FOR N=1 TO 20
120 PRINT AT 0,N;A$(1)
130 PRINT TAB N;A$(2)
140 PRINT TAB N;A$(3)
150 PRINT TAB N;A$(4)
170 PAUSE 20
180 IF INKEY$<>"" AND N=11 THEN
GOTO 500
190 NEXT N
200 CLS
210 NEXT F
220 CLS
280 PRINT "YOU GOT ";S
290 STOP
500 PRINT AT 2,N;A$(5)
510 LET S=S+1
520 PAUSE 60
530 CLS
540 LET N=22
550 NEXT N
560 GOTO 210
```

NIM

This game, based on one which was featured in the film "Last Year at Marienbad" fits a 1K machine. There are between 15 and 23 objects on the screen at the start of the game, and you and the computer take it in turns to take one,

two or three of these away. The player who removes the last one loses.

```
  10 LET M=0
  20 LET E=0
  30 LET Z=15+INT (RND*10)
  40 IF 2*INT (Z/2)=Z THEN GOTO
30
  50 LET H=3
  60 IF E>0 THEN PRINT AT 7,0;"Y
OU TOOK ";CHR$ (E+156),"I TOOK "
;CHR$ (Q+156)
  65 PRINT
  70 FOR K=1 TO Z
  80 PRINT K;" ▀";
  90 IF RND>.6 THEN PRINT
 100 NEXT K
 110 INPUT E
 120 LET Z=Z-E
 130 IF Z=0 THEN PRINT ,"I WIN";
W
 150 LET Q=Z-1-INT ((Z-1)/(H+1))
*(H+1)+INT (RND*4)
 160 IF Q>Z OR Q<1 OR Q>3 THEN G
OTO 150
 170 LET Z=Z-Q
 180 IF Z=0 THEN PRINT ,"YOU WIN
";W
 200 CLS
 210 GOTO 60
```

BUGBITE

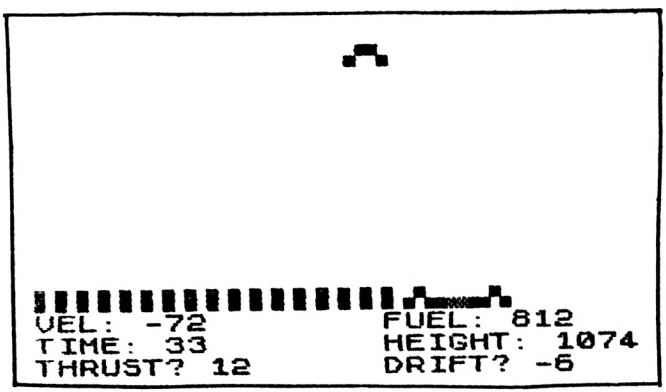
You and the ZX81/new ROM ZX80, take it in turns to roll a four-sided die to build up a picture of a BUGBITE. The 1K computer does all the work (and often wins). The ZX81's BUGBITE is on the right, yours is on the left. A one gets you a head, two and three get you legs, and four gets you the body. You need to roll the numbers in the right order (i.e. one first, then two and so on) to complete your BUGBITE.

```
 10 LET A$="  ) ) )"
 20 LET B$="  ▄▄▄▄▄ "
 30 LET C$=" █▀▀▀▀█ "
 40 LET D$=" ▒█▀▀▀ "
 50 LET E$="▒▒█▄█ "
 60 DIM M(2)
 65 LET F$="   YOU    I"
 70 FOR N=1 TO 2
 80 LET D=INT (RND*4)+1
```

```
  90 PRINT AT 0,0;F$(N*3 TO N*3+
2);" HAVE A ";D
 100 PAUSE 100
 110 IF D-PI/PI=M(N) THEN LET M(
N)=D
 120 IF M(N)<PI THEN GOTO 150
 130 PRINT AT 3,N**PI;B$
 140 PRINT TAB N**PI;C$
 150 IF M(N)<3 THEN GOTO 170
 160 PRINT AT 2,N**PI;A$
 170 IF M(N)<2 THEN GOTO 190
 180 PRINT AT 5,N**PI;A$
 190 IF M(N)=PI-PI THEN GOTO 220
 200 PRINT AT PI,N**PI+4;D$
 210 PRINT TAB N**PI+4;E$
 220 IF M(N)>PI THEN PRINT F$(N*
3 TO N*3+2);" WON.";Z
 230 NEXT N
 240 GOTO 70
```

MOON BASE

In this lunar lander game (which needs more than 1K), you have to try and land your little craft on the base at the end of the line of upright posts on the moon's surface. You have



two inputs: THRUST which controls your descent; and DRIFT which controls your lateral movement. The game ends if (a) you crash land; you'll be awarded a rating). The printout shows a sample game in action.

```
   1 GOTO 42
   2 IF H<50 AND V<20 AND V>-15
AND ABS (Z-M)<5 THEN GOTO 37
   3 IF H>1750 THEN GOTO 40
   4 RETURN
   5 LET A=A/7
   6 LET T=T+4+INT (RND*2+1)
   7 LET V=V+A**3-12-3*RND
   8 LET H=H+V-20+10*RND
   9 LET F=F-(ABS (A)+ABS (B/5)*
6*RND)
  10 GOSUB 2
  11 IF H<20 OR F<5 THEN GOTO 35
  12 LET U=H/100
  13 LET Z=Z+B/2+2-RND*3
  16 PRINT AT W,X;"   "
  17 PRINT AT 16-U,Z+1;"▄▀▀▄"
  18 LET W=16-U
  19 LET X=Z+1
  21 PRINT AT 16,0;"█████████████
██████████"( TO M-1);"▄▄▄▄▄▄▄"
  28 PRINT AT 17,0;"VEL: ";INT V
;"   ","FUEL: ";INT F;"   ","TIME:
 ";INT T;"   ","HEIGHT: ";INT H;"
 "
  29 PRINT "THRUST? ";
  30 INPUT A
  31 PRINT A;"   ","DRIFT? ";
  32 INPUT B
  33 PRINT B
  34 GOTO 5
  35 PRINT "CRASH SPEED ";ABS V
;"  ";
  36 GOTO 35
  38 PRINT "SUCCESSFUL LANDING▒R
ATING ";100*(30-INT ABS V)+INT V
;"▒";
  39 GOTO 38
  40 PRINT "YOU HAVE REACHED ESC
APE VELOCITY"
  41 GOTO 40
  42 LET H=1450
  43 LET F=827+50*RND
  44 LET T=0
  45 LET Z=15*RND
  46 LET A=1
  47 LET B=0
  48 LET M=19*RND
  49 LET V=0
  50 LET W=0
  51 LET X=0
  70 GOTO 6
```

SHOWOFF

Run this great little program the first time you want to impress your friends with your new ROM ZX80 or ZX81. Delete the PAUSE lines, except for 180, if running it in SLOW. This needs 1K.

```
  10 LET A$="HELLO I AM A COMPUT
ER"
  20 GOSUB 500
  30 LET A$="YOU CAN CALL ME ZED
DY"
  40 GOSUB 500
  50 LET A$="I CAN DO SUMS...."
  60 GOSUB 500
  70 LET A$="TYPE IN 1 AND I WIL
L DO IT"
  80 GOSUB 500
  90 INPUT A$
 100 LET A$="THE ANSWER IS "+STR
$ VAL A$
 110 GOSUB 500
 120 LET A$="ILL HAVE TO GO NOW.
BYE..."
 130 GOSUB 500
 140 CLS
 150 PRINT AT 9,13;"
 160 PRINT TAB 13;"
 170 PRINT TAB 13;"
 180 PAUSE 9999
 190 RUN
 500 CLS
 510 FOR N=1 TO LEN A$
 520 PRINT AT 10,N;A$(N);"*"
 530 PAUSE 20
 540 NEXT N
 550 PAUSE 50
 560 RETURN
```

MUSIC

The music produced by this routine is pretty horrible but gives you an outline to follow. The original idea came from L. D. Tanner of Waddon, Croydon, a member of the National ZX80 and ZX81 Users' Club. Put a transistor radio near your computer to hear the "music". You may be able

to hear it through your TV if you turn the sound up, but you may have to tune it slightly away from the "best picture" position to hear the music clearly.

```
1    LET K = 100
2    FOR G = 1 TO 2
3    FOR A = 1 TO K
4    GOSUB 30
5    NEXT A
6    FOR A = 1 TO K
7    GOSUB 30
8    NEXT A
9    FOR A = 1 TO K
10   LET J = J + 1
11   GOSUB 30
12   NEXT A
13   PAUSE 5
14   FOR S = 1 TO 3
15   FOR A = 1 TO 17
16   GOSUB 30
17   NEXT A
18   NEXT S
19   NEXT G
20   PAUSE 5
21   FOR A = 1 TO 50
22   GOSUB 30
23   NEXT A
24   FOR A = 1 TO K
25   GOSUB 30
26   LET J = J + 1
27   NEXT A
28   PAUSE 10
29   RUN
30   SLOW
31   FAST
32   LET J = Ø
33   RETURN
```

MENACE

Menacing aliens chug-a-lug up the screen towards you (an inverse V). You move right and left with the "5" and "8" keys, and the game continues until an alien hits you. This must be run in FAST. It takes just 1K.

```
1 LET P=16
2 LET S=Ø
3 PRINT AT Ø,P;"▓"
4 PAUSE 13.
```

```
   5 SCROLL
   6 PRINT AT 16,RND*29;"■■■";AT
 0,31;" "
   7 IF PEEK (P+16743)>0 THEN GO
TO 11
   8 LET P=P+(INKEY$="8")-(INKEY
$="5")
   9 LET S=S+1
  10 GOTO 3
  11 SCROLL
  12 PRINT S
```

MAGIC SQUARE

The computer (a ZX81 or new ROM ZX80 with more than 1K) generates a magic square, in which the numbers horizontally, diagonally and vertically add up to the same total. A zero will be printed in three of the squares, and you have to work out which numbers should take their places. Enter any guess, and the computer will check each of the zero places to see if this number should take its place. The computer automatically knows when you have completed the magic square, and terminates the game.

```
   5 DIM A(9)
  10 DIM B(9)
  15 LET W=-99
  20 LET A=INT (RND*9)+1
  25 LET J=0
  30 LET B=INT (RND*9)+1
  40 LET C=INT (RND*9)+1
  50 IF A=B OR A=B OR A=C OR B=C
THEN GOTO 30
  60 LET A(1)=A+B
  70 LET A(2)=A-(B+C)
  80 LET A(3)=A+C
  90 LET A(4)=A-B+C
 100 LET A(5)=A
 110 LET A(6)=A+B-C
 120 LET A(7)=A-C
 130 LET A(8)=A+B+C
 140 LET A(9)=A-B
 150 FOR Z=1 TO 9
 160 LET B(Z)=A(Z)
 170 NEXT Z
 180 LET K=ABS A
 190 LET B(K)=0
 200 LET K=ABS B
 210 LET B(K)=0
```

```
 220  LET K =ABS C
 230  LET B(K)= Ø
 235 LET J=J+1
 240 PRINT AT 0,0;"GUESS NO. ";J
 245 PRINT
 247 PRINT
 250 FOR Z=1 TO 9
 260 PRINT B(Z);"    ";
 270 IF Z=3 OR Z=6 THEN PRINT
 275 IF Z=3 OR Z=6 THEN PRINT
 280 NEXT Z
 282 PRINT
 285 IF M=9 THEN PRINT "YOU HAVE
 SOLVED IT"
 287 IF M=9 THEN PRINT "
     ";Ø
 288 PRINT
 289 PRINT "YOU HAVE ";M;" RIGHT
"
 290 INPUT W
 295 LET M=Ø
 300 FOR Z=1 TO 9
 305 IF W=-99 THEN GOTO 320
 310 IF A(Z)=W THEN LET B(Z)=W
 320 IF B(Z)<>Ø THEN LET M=M+1
 330 NEXT Z
 340 GOTO 235
```

TWENTY-ONE

You and the 1K ZX81 take it in turns to roll a die, trying to get a total as close as possible to, but not exceeding, 21. If you have more memory, add a facility for multiple games, with an accumulating score.

```
   10 LET H=0
   20 LET C=0
   30 PRINT "1 TO ROLL, 2 TO STAN
D"
   35 INPUT A
   40 IF A=2 THEN GOTO 90
   50 LET H=H+INT (RND*6)+1
   60 GOSUB 260
   70 PRINT AT 4,7;"[illegible] ";H
   80 GOTO 35
   90 IF C>H AND C<22 OR C>21 OR
H>21 OR H=21 AND C=21 THEN GOTO
140
  100 LET C=C+INT (RND*6)+1
  110 GOSUB 260
  120 PRINT AT 8,8;"[illegible] ";C
  130 GOTO 90
  140 PRINT AT 11,8;
  150 GOSUB 260
  160 GOSUB 260
  170 IF H=C OR H>21 AND C>21 THE
N GOTO 240
  180 IF (C>H OR H>21) AND C<22 T
HEN PRINT "[illegible]";
  190 IF (C<H OR C>21) AND H<22 T
HEN PRINT "[illegible]";
  200 PRINT "[illegible]"
  210 GOSUB 260
  215 GOSUB 260
  220 CLS
  230 RUN
  240 PRINT "[illegible]"
  250 STOP
  260 FOR E=1 TO 60
  270 NEXT E
  280 RETURN
```

TOWER

Pick yourself a tower, 1, 2 or 3, then stand back and let the ZX81 build them for you, and see if yours is finished first.

The routine in lines 120/130 seeks to increase the randomness of the random number generator.

```
10   LET D = 30
20   LET A = 5
30   LET B = A
40   LET C = A
50   PLOT 10,A
60   PLOT 30,B
70   PLOT 50,C
80   LET A = A + RND
90   LET B = B + RND
100  LET C = C + RND
110  IF(A > D OR B > D OR C > D) THEN GOTO 140
120  LET X = INT(RND*6)+1
130  GOTO 50*(X < 4) + 60*(X > 3 AND A < 6) +
     70*(X = 6)
140  PRINT   (A > B AND A >C) + 2*(B > A + B > C)
     + 3*(C > A AND C > B);" WINS"
```

HANGPERSON

This is HANGMAN in reverse. You think of a word, and the computer tries to guess it. When you RUN this program, the computer will first ask you how many letters there are in the word.

The ZX81 will then think of a letter. If this letter is in your word, type the numberof the letter in the word. That is, if your word is APPLE and the computer guesses E, respond by typing 5. If the letter is wrong, input Ø. After a correct letter, the computer will leave it there until you typeØ, to allow for double letters. So, if it thought of P, you'd respond with 2, then NEWLINE, then 3, then NEWLINE, then Ø. This game is great fun to play, and you'll find the long-suffering ZX81 has a much better chance of guessing your word within its 10 goes if you think of a long word.

```
  10 REM ETAONRISHDLFCMUGYPWBJKQ
XVZ
  20 LET L=10
  30 PRINT "LENGTH OF WORD?"
  40 INPUT N
```

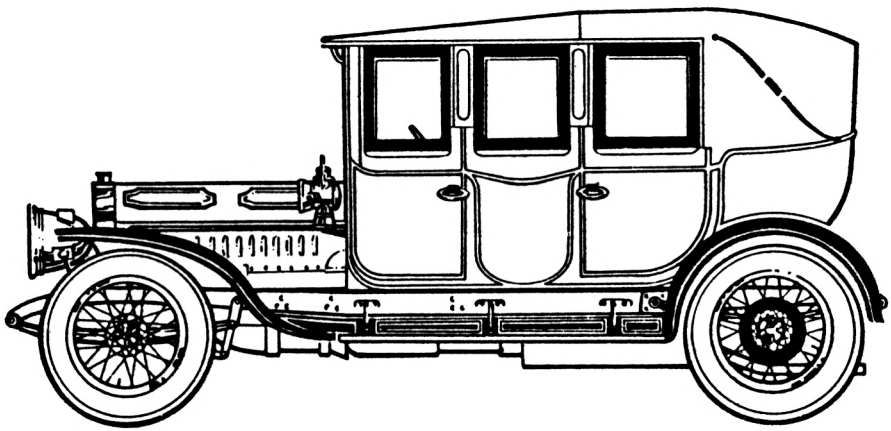
```
  50 CLS
  80 DIM A(26)
  90 DIM G(N)
 100 DIM G(N)
 110 FOR Z=1 TO 26
 120 LET A(Z)=PEEK (16513+Z)
 130 IF Z<N+1 THEN LET G(Z)=4
 140 NEXT Z
 150 LET Z=INT (RND*3)+1
 160 LET A$=CHR$ A(Z)
 170 FOR J=Z TO 25
 180 LET A(J)=A(J+1)
 190 NEXT J
 200 LET A=0
 210 PRINT AT 3,4;
 230 FOR Z=1 TO N
 240 PRINT CHR$ G(Z);
 250 NEXT Z
 260 PRINT
 270 PRINT
 280 PRINT TAB 8;"LIVES: ";L;" "
TAB 10;"I GUESS ";A$
 300 INPUT B
 310 IF B=0 THEN GOTO 350
 320 LET A=1
 330 LET G(B)=CODE A$
 340 GOTO 210
 350 LET F=0
 360 FOR Z=1 TO N
 370 IF G(Z)=4 THEN LET F=1
 380 NEXT Z
 390 IF F=0 THEN PRINT TAB 8;"I
WIN";W
 410 IF A=0 THEN LET L=L-1
 420 IF L>0 THEN GOTO 150
 430 PRINT TAB 8;"YOU WIN"
```

AVOID

In this 1K ZX81 game, a grey blob moves back and forth under your control. The key marked "1" moves you left, and "0" moves you right. The aim is to avoid the moving black squares. If you hit one, the black square turns into your score. Any score over 95 is good.

```
5    LET K = 0
10   POKE 16418,8     (this line changes the line
                        from which SCROLL operates)
20   LET A = 5
30   LET B = A
40   SCROLL
50   PRINT AT A,B;"graphic H"
60   LET C = A
```

```
70   LET D = B
75   IF INKEY$ = "" THEN GOTO 90
80   LET B = B - (B > 1 AND INKEY$ = "1") +
                 (B < 19 AND INKEY$ = "0")
90   PRINT AT C,D;"single space"
100  PRINT AT 8,INT(RND*20);"inverse space"
110  LET K = K + 1
120  PRINT AT 6,D;
130  IF PEEK(PEEK 16398 + 256*PEEK 16399)<>128
     THEN GOTO 40
140  PRINT K
```



BOMBER

A tiny plane flies overhead. You press any key to fire at the target (a graphic H). The line across the top of the screen gets shorter and shorter, and you can keep playing until the line vanishes. You get a satisfying BOOM if you hit the target. This game is 1K for ZX81.

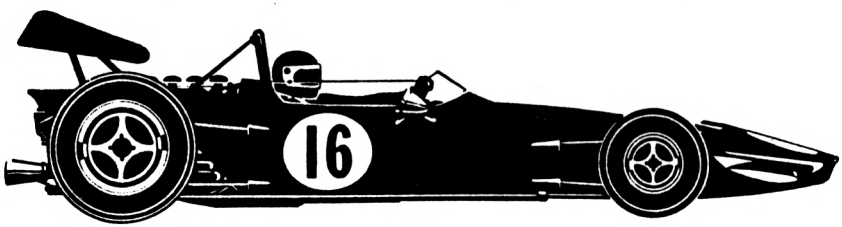
```
 10 LET T=16
 15 LET B=0
 16 LET D=1
 20 PRINT "
"
 30 LET F=64
 40 FOR N=1 TO 30
 50 PRINT AT 1,N-1;" ";AT 8,T
" ";AT D,N;"▚"
 60 PAUSE 15
 70 LET B=B OR INKEY$<>""
```

```
 80 IF B THEN GOTO 200
 90 LET T=T+INT (RND*3)-1
100 LET F=F-1
110 UNPLOT F,43
120 IF NOT F THEN STOP
130 NEXT N
140 PRINT AT 1,30;"  "
150 GOTO 40
200 PRINT AT D,N;" "
210 LET D=D+1
220 IF D=8 AND N=T THEN PRINT T
AB N;"BOOM",Z
230 IF D<8 THEN GOTO 90
240 LET B=0
250 LET D=1
260 GOTO 90
```

RALLY

In this 1K ZX81 game, you are an inverse V, trying to negotiate a very difficult track, strewn with obstacles. The score clocks up beside the track throughout the game. The higher this score becomes, the better.

Anything over 183 is a great score. Run it in FAST on a ZX81, and use the "5" and "8" to control your car.




```
 10 LET P=16770
 20 LET S=0
 30 SCROLL
 40 LET B$="                "
 50 LET B$(RND*7+2)="  "
 55 LET B$(RND*7+2)="  "
 60 PRINT B$;AT 0,10;S
 70 IF PEEK P<>0 THEN STOP
 80 POKE P,187
 90 PAUSE 20
100 LET S=S+1
110 LET P=P+(INKEY$="8")-(INKEY
$="5")
120 GOTO 30
```

LIFE

The game simulates the birth, growth and death of a cell colony, producing fascinating effects as it does so. The cells live on a grid (in these versions it is a 10 x 10 grid) and are born, live or die according to Conways rules:

.Each cell on the grid has eight neighbours

.Every cell with two or three neighbours survives to the next generation

.If there are three, and only three, neighbouring cells, a new cell is born

.Any cell with four or more neighbours dies from overpopulation

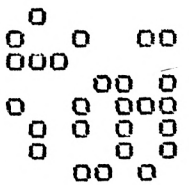
```
  5 LET G=0
  7 RAND
 10 DIM A(10,10)
 20 DIM B(10,10)
 30 FOR X=2 TO 9
 40 FOR Y=2 TO 9
 50 IF RND>.35 THEN LET A(X,Y)=
1
 60 LET B(X,Y)=A(X,Y)
 70 NEXT Y
 80 NEXT X
 90 GOSUB 1000
 95 LET G=G+1
100 FOR X=2 TO 9
110 FOR Y=2 TO 9
120 LET C=0
130 IF A(X-1,Y-1)=1 THEN LET C=
C+1
140 IF A(X-1,Y)=1 THEN LET C=C+
1
150 IF A(X-1,Y+1)=1 THEN LET C=
C+1
160 IF A(X,Y-1)=1 THEN LET C=C+
1
170 IF A(X,Y+1)=1 THEN LET C=C+
1
```

```
 180 IF A(X+1,Y-1)=1 THEN LET C=
C+1
 190 IF A(X+1,Y)=1 THEN LET C=C+
1
 200 IF A(X+1,Y+1)=1 THEN LET C=
C+1
 210 IF A(X,Y)=1 AND C<>3 AND C<
>2 THEN LET B(X,Y)=0
 220 IF A(X,Y)=0 AND C=3 THEN LE
T B(X,Y)=1
 230 NEXT Y
 240 NEXT X
 250 GOTO 90
1000 PRINT AT 3,9;"■GENERATION■"
;G
1001 FOR X=1 TO 10
1010 FOR Y=1 TO 10
1015 LET A(X,Y)=B(X,Y)
1020 IF A(X,Y)=1 THEN PRINT AT X
+4,Y+10;"O"
1030 IF A(X,Y)=0 THEN PRINT AT X
+4,Y+10;" "
1040 NEXT Y
1050 NEXT X
1060 RETURN
```

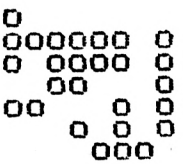
```
■GENERATION■0

  OOO   OO
O       O
 O OOO O
O    O OO
   OOO O
OOO O
O O   OOO
OOOO OOO
```

```
■GENERATION■1

  OO    OO
O
OO OO   O
 O       O
O O O OO
O    O   O
          O
O OOOO O
```

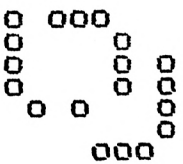
GENERATION 2



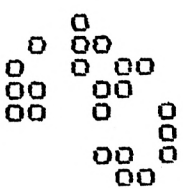
GENERATION 3



GENERATION 4



GENERATION 5



```
GENERATION 13

   O
  OOO
 OOOOO
OO
OO
        OO
       OOO

GENERATION 14

    OOO
  O   O
O     O
   OO
OO
        O O
        O O
         O
```

MIRROR LIFE

```
  5 LET G=0
  7 RAND
 10 DIM A(10,10)
 20 DIM B(10,10)
 30 FOR X=2 TO 9
 40 FOR Y=2 TO 9
 50 IF RND>.45 THEN LET A(X,Y)=
1
 60 LET B(X,Y)=A(X,Y)
 70 NEXT Y
 80 NEXT X
 90 GOSUB 1000
 92 FAST
 95 LET G=G+1
100 FOR X=2 TO 9
110 FOR Y=2 TO 9
120 LET C=0
130 IF A(X-1,Y-1)=1 THEN LET C=
C+1
140 IF A(X-1,Y)=1 THEN LET C=C+
1
150 IF A(X-1,Y+1)=1 THEN LET C=
C+1
160 IF A(X,Y-1)=1 THEN LET C=C+
1
170 IF A(X,Y+1)=1 THEN LET C=C+
1
```

```
 180 IF A(X+1,Y-1)=1 THEN LET C=
C+1
 190 IF A(X+1,Y)=1 THEN LET C=C+
1
 200 IF A(X+1,Y+1)=1 THEN LET C=
C+1
 210 IF A(X,Y)=1 AND C<>3 AND C<
>2 THEN LET B(X,Y)=0
 220 IF A(X,Y)=0 AND C=3 THEN LE
T B(X,Y)=1
 230 NEXT Y
 240 NEXT X
 250 GOTO 90
1000 PRINT AT 3,4;"GENERATION ";
G
1001 FOR X=1 TO 10
1002 FOR Y=1 TO 10
1003 SLOW
1015 LET A(X,Y)=B(X,Y)
1020 IF A(X,Y)=1 THEN PRINT AT X
+4,Y+10;"O"
1025 IF A(X,Y)=1 THEN PRINT AT 1
4-X,12-Y;"O"
1030 IF A(X,Y)=0 THEN PRINT AT X
+4,Y+10;" "
1035 IF A(X,Y)=0 THEN PRINT AT 1
4-X,12-Y;" "
1040 NEXT Y
1050 NEXT X
1055 COPY
1060 RETURN
```

```
     GENERATION 0

     OO O  O
    O O  O O OO  OO
    O OOOOO   OO
    OO O  OO  OO O
    OOO O        O OOO
       O OO  OO  O OO
         OO   OOOOO O
      OO  OO O O  O O
             O  O OO

     GENERATION 1

     OOO  O
    O      O OOO
    O           OO
          OO  O  OOO
    O   O  O O  O   O
     OOO  O  OO
      OO              O
         OOO O        O
               O  OOO
```

```
 GENERATION 2

 OO
O O       O
      OO O OOO O
      OO    O  O
 OOO O O O O OOO
 O  O    OO
 O OOO O OO
      O       O O
              OO
```

```
 GENERATION 13

OO
  O       OOOOOO
OOO      O OOOOO
 OO      OO
      OO OO
      OO      OO
 OOOOO O      OOO
 OOOOOO       O
               OO
```

4-IN-A-ROW

As you can see from the screen printout, this game is played on a 10 x 10 board. You and another human take it in turns to indicate which row (A to J) you want to use. A piece appears, an X or an O, at the lowest vacant slot in that row. The aim is to get four in a row, in any direction. In the sample game, X has just won, with a diagonal row starting from C and going up to F. This program needs more than 1K.


```
  10 DIM A(100)
  20 FOR J=1 TO 2
  40 LET C=52*(J=1)+51*(J=2)
  60 PRINT AT 5,7;"■ABCDEFGHIJ"
  70 PRINT TAB 7;"■";
  80 FOR B=1 TO 100
  90 PRINT CHR$ A(B);
 100 IF 10*INT (B/10)=B THEN PRI
NT TAB 7;"■";
 110 NEXT B
 120 PRINT AT 16,8;"ABCDEFGHIJ"
 130 PRINT AT 3,8;"PLAYER ";J
 140 INPUT Z$
 145 IF Z$="S" THEN STOP
 150 LET M=CODE Z$-37
 160 FOR B=M TO M+90 STEP 10
 170 IF A(B)=0 THEN NEXT B
 180 LET A(B-10)=C
 190 NEXT J
 200 GOTO 20
```

SANDOWN

This is a race between three numbers, 1, 2, and 3. Line 130 is based on the fact that the ZX81 evaluates TRUE as 1 and FALSE as 0.

```
5    LET D = 30
10   LET A = 0
20   LET B = A
30   LET C = A
40   LET K = 150
50   PRINT AT 2,A;"1"
60   PRINT AT 4,B;"2"
70   PRINT AT 6,C;"3"
74   GOSUB K
```

```
78   PRINT AT 2,A;"space"
80   LET A = A + RND
82   GOSUB K
85   PRINT AT 4,B;"space"
90   LET B = B + RND
100  PRINT AT 6,C;"space"
105  LET C = C + RND
110  GOSUB K
120  GOTO 50
130  PRINT   (A> C  AND  A> B) + 2*(B >A AND B> C)
     + 3*(C >A AND C > B);" WINS"
140  STOP
150  IF (A> D OR B> D  OR C>  D) THEN GOTO 130
160  RETURN
```

DEMON

This program, which needs a ZX81 with at least 4K, is based partly on pawn moves in chess, and partly on draughts.

DEMON is played on a six by six grid of dots. You have six men each at the start of the game. You are the X's and the ZX81 is the 0's.

You move in a diagonal direction only (as in draughts), but you can move forwards or backwards . You capture an

```
ZX81 0                HUMAN 0

. X . X . X A
X . X . X . B
. . . . . . C
. . . . 0 . D
. 0 . 0 . . E
0 . 0 . 0 . F
1 2 3 4 5 6
YOUR MOVE?
```

opponent's piece by landing on top of it. The first player to capture four of the opponent's pieces wins.

The ZX81 always has first move. You move by entering the square LETTER and NUMBER of the piece you want to

move, then the letter and number of the square you're moving to, and THEN press NEWLINE. That is, you enter the "square from" and the "square to" before you press NEWLINE.

You'll see that the board is stored in the first REM statement, and the computer makes decisions by PEEKing into this REM statement. Lines 21 to 23 rePOKE the board into the first REM statement at the start of each new game. This game can be adapted fairly easily for the ZX80, but you'll have to add a few INPUT A$'s and CLS to control the display.

```
  10 REM .X.X.XX.X.X.............
..0.0.00.0.0.123456
  20 REM .X.X.XX.X.X.............
..0.0.00.0.0.123456
  21 FOR K=16562 TO 16598
  22 POKE (K-48),PEEK K
  23 NEXT K
  24 LET N=0
  25 LET Z=16513
  26 LET S=0
  27 DIM B(4)
  28 LET B(1)=5
  29 LET B(2)=7
  30 LET B(3)=-5
  31 LET B(4)=-7
  32 SLOW
  33 LET K=0
  35 GOSUB 68
  37 FAST
  40 GOTO 330
  68 PRINT AT 5,0;"ZX81 ";N,"HUM
AN ";S
  69 PRINT
  70 FOR C=1 TO 36
  71 PRINT CHR$ PEEK (Z+C);" ";
  73 IF 6*(INT (C/6))=C THEN PRI
NT CHR$ (C/6+37)
  75 NEXT C
  77 PRINT "1 2 3 4 5 6"
  78 FOR H=1 TO 30
  80 NEXT H
  90 RETURN
 120 GOSUB 68
 125 SLOW
 130 PRINT "YOUR MOVE?"
```

```
 132 INPUT A$
 136 LET D=6*(CODE A$(1)-38)+COD
E (A$(2))-28
 140 LET E=6*(CODE A$(3)-38)+COD
E (A$(4))-28
 150 IF PEEK (E+Z)=52 THEN LET S
=S+1
 155 POKE (D+Z),27
 160 POKE (E+Z),61
 170 GOSUB 68
 175 FAST
 180 IF S=4 THEN GOTO 410
 190 LET K=0
 250 FOR F=1 TO 36
 260 FOR M=1 TO 4
 270 IF PEEK (F+Z)=52 AND PEEK (
F+Z+B(M))=61 THEN LET N=N+1
 275 IF PEEK (F+Z)=52 AND PEEK (
F+Z+B(M))=61 THEN GOTO 430
 290 NEXT M
 300 NEXT F
 330 LET M=1
 340 LET F=INT (RND*36)+1
 342 LET K=K+1
 343 PRINT AT 0,0;K
 350 IF PEEK (F+Z)<>52 THEN GOTO
 340
 352 IF RND>.2 AND M<3 THEN LET
M=4-INT RND
 355 IF (F=6 OR F=18 OR F=30) AN
D (M=2 OR M=3) THEN GOTO 370
 357 IF (F=31 OR F=19 OR F=7) AN
D (M=1 OR M=4) THEN GOTO 370
 360 IF PEEK (F+Z+B(M))=27 AND P
EEK (F+Z+2*B(M))<>61 THEN GOTO 4
30
 370 LET M=M+1
 375 IF M<5 THEN GOTO 355
 400 IF K<100 THEN GOTO 330
 410 PRINT "YOU WIN"
 420 STOP
 430 SLOW
 435 POKE (F+Z+B(M)),128
 437 GOSUB 68
 440 IF N=4 THEN GOTO 500
 460 POKE (F+Z),27
 465 POKE (F+Z+B(M)),52
 470 GOTO 120
 500 PRINT "I WIN"
```

DEMONSTRATIONS AND SUBROUTINES

Here are three short programs which you may like to incorporate into longer programs you write.

TRIANGLES

You enter any number from 2 to 15 and the clever 1K ZX81 draws a triangle with that base number, counting the number of points within the triangle.

```
 10 PRINT "BASE OF TRIANGLE"
 20 INPUT B
 30 CLS
 40 PRINT "BASE ";B,"TOTAL "
 50 LET T=0
 60 LET S=0
 70 LET L=20
 80 LET T=T+B
 90 FOR N=S TO S+B*2-2 STEP 2
100 PRINT AT L,N;"O";
110 NEXT N
120 LET L=L-1
130 LET B=B-1
140 LET S=S+1
150 IF B>0 THEN GOTO 80
160 PRINT AT 0,22;T
```

MISSILE

When you press "0", you'll see your missile fire towards the target. You should be able to build a good game around this routine.

```
   1 PRINT AT 9,15;"▚ ▞";TAB 16;
"X";TAB 15;"▞ ▚"
   2 IF INKEY$="0" THEN GOSUB 10
   4 GOTO 2
  10 FOR N=2 TO 21
  11 PLOT 10+N,N
```

```
12 PLOT 55-N,N
13 UNPLOT 8+N,N-2
14 UNPLOT 57-N,N-2
16 NEXT N
17 RETURN
```

SOLID SINE

This 1K ZX81/new ROM ZX80 program is a very effective demonstration of the graphics capability of your computer. It plots a "solid" sine wave.

```
 1 FOR X=0 TO 63
 2 LET Y=20*SIN (X/32*PI)
 3 IF Y=0 THEN GOTO 7
 4 FOR N=0 TO Y STEP SGN Y
 5 PLOT X,N+22
 6 NEXT N
 7 NEXT X
 8 PRINT AT 10,0;"----------
----------"
```

HOW LONG HAVE I GOT?

This program asks you a number of questions, responds to them with wise comments, and then tells you how long you will live -- based on statistics. Do not take the results too seriously. The words in inverse graphics are:

180 MALE FEMALE
220 ONE
320 MARRIED
590 DRINKING
640 DO YOU SMOKE (Y OR N)?
680 10 - 20 CIGARETTES - A
690 20 - 30 CIGARETTES - B
700 MORE THAN 30 A DAY - C
710 A PIPE OR CIGAR - D
800 ILL

```
  10 LET H$=" "
  20 LET N$=" "
  30 RAND
  40 PRINT "    LETS HAVE A LOOK
AT YOUR"
  50 PRINT TAB (8);"LIFE EXPECTA
NCY"
  60 PRINT
  90 PRINT "FIRST, WHAT IS YOUR
NAME?"
 100 INPUT T$
 110 CLS
 120 PRINT
 140 PRINT "OK, ";T$;", WHAT YEA
R"
 150 PRINT "WERE YOU BORN? (GIVE
 ANSWER          IN THE FORM -196
4)"
 160 INPUT A
 170 GOSUB 2000
 180 PRINT "ARE YOU [redacted] (1) OR
[redacted] (2)?"
 190 INPUT B
 200 GOSUB 2000
 210 PRINT "WHICH AGE GROUP ARE
YOU IN?"
 220 PRINT TAB (8);"INPUT [redacted] LE
TTER"
 230 PRINT " 5 TO 25 - A","26 TO
 40 - B","41 TO 50 - C","51 TO 6
0 - D","61 TO 65 - E","66 TO 70
 - F","71 TO 75 - G"," OVER 75 -
H"
 240 INPUT A$
 250 GOSUB 2000
 260 PRINT "DID/HAS YOUR FATHER
LIVE(D) PAST        70 (Y OR N)"
 270 INPUT B$
```

```
 280 GOSUB 2000
 290 PRINT "DID/HAS YOUR MOTHER
LIVE(D) PAST          70 (Y OR N)"
 300 INPUT C$
 310 GOSUB 2000
 320 PRINT "ARE YOU [redacted] (Y O
R N)?"
 330 INPUT D$
 340 GOSUB 2000
 350 PRINT TAB (6);"WHERE DO YOU
 LIVE?","      A SMALL TOWN - A"
,"            A CITY - B"
 360 INPUT E$
 370 GOSUB 2000
 380 PRINT "HAVE YOU BEEN RICH O
R POOR FOR  MOST OF YOUR LIFE (Y
 OR N)?"
 390 INPUT F$
 400 IF CODE (A$)<40 THEN GOTO 4
70
 410 GOSUB 2000
 420 PRINT "ARE YOU OVERWEIGHT (
Y OR N)?"
 430 INPUT G$
 435 PRINT
 440 IF CODE (G$)<>62 THEN GOTO
470
 450 PRINT TAB (12);"A LITTLE -
A"
 452 PRINT TAB (7);"A MODERATE A
MOUNT - B"
 454 PRINT TAB (10);"QUITE A BIT
 - C"
 460 INPUT H$
 470 GOSUB 2000
 480 PRINT "EXERCISE..."
 485 PRINT "[redacted]"
 490 PRINT
 500 PRINT
 510 PRINT
 520 PRINT "HOW MUCH EXERCISE DO
 YOU GET?"
 522 PRINT TAB (8);"VERY LITTLE
 - A"
 524 PRINT TAB (8);"A MODERATE A
MOUNT - B"
 526 PRINT TAB (8);"A LOT - C"
 530 INPUT J$
 540 GOSUB 2000
 550 PRINT "ARE YOU USUALLY:-"
 552 PRINT TAB (4);"GOOD-NATURED
 AND PLACID - A"
 554 PRINT TAB (7);"TENSE AND NE
RVOUS - B"
 556 PRINT TAB (11);"IN BETWEEN
 - C"
 570 INPUT K$
```

```
  580 GOSUB 2000
  585 PRINT "                    "
  590 PRINT "      [redacted]"
  592 PRINT "      [redacted]"
  600 PRINT
  610 PRINT "HOW OFTEN DO YOU DRI
NK: -"
  611 PRINT "      RARELY OR NEVER
 - A"
  612 PRINT "      OCCASIONALLY - B
"
  613 PRINT "      REGULARLY (MODER
ATELY) - C"
  614 PRINT "      REGULARLY (HEAVI
LY)  - D"
  615 PRINT "      REGULARLY (VERY
HEAVILY)  - E"
  620 INPUT L$
  630 GOSUB 2000
  640 PRINT "[redacted]
[redacted]"
  650 INPUT M$
  660 CLS
  670 IF CODE (M$)<>62 THEN GOTO
730
  675 PRINT
  676 PRINT
  677 PRINT
  680 PRINT "[redacted]
[redacted]"
  690 PRINT "[redacted]
[redacted]"
  700 PRINT "      [redacted]
[redacted]"
  710 PRINT "      [redacted]
[redacted]"
  720 INPUT N$
  730 GOSUB 2000
  740 PRINT "  DO YOU VISIT A DEN
TIST AT"
  750 PRINT " LEAST TWICE A YEAR
(Y OR N)?"
  755 INPUT P$
  760 GOSUB 2000
  770 PRINT "DO YOU HAVE REGULAR
MEDICAL              CHECKUPS (Y OR
N)?"
  780 INPUT Q$
  790 GOSUB 2000
  800 PRINT "ARE YOU OFTEN [redacted]
 Y OR N)?"
  810 INPUT R$
  820 LET L=48*(A<1911)+52*(A>191
0 AND A<1921)+59*(A>1920 AND A<1
931)+61*(A>1930 AND A<1941)+65*(
A>1940 AND A<1951)+67*(A>1950 AN
D A<1961)+68*(A>1960)
```

```
 830 IF B=2 THEN LET L=51*(L=48)
+56*(L=52)-62*(L=59)+67*(L=61)+7
1*(L=65)+74*(L=67)+75*(L=68)
 835 LET L4=1990-A
 840 LET V=CODE (A$)-37
 850 LET L1=3*(V=1)+5*(V=3 OR V=
6)+7*(V=4)+8*(V=5)+10*(V=6)+12*(
V=7)
 860 LET L=L+L1
 870 LET L=L+(CODE (B$)=62)
 880 LET L=L+(CODE (C$)=62)
 890 LET L=L+3*(CODE (D$)=62)
 900 LET L=L+4*(CODE (E$)=38)-2*
(CODE (E$)=39)
 910 LET L=L-3*(CODE (F$)=62)
 920 LET L=L-(CODE (H$)=38)-3*(C
ODE (H$)=39)-5*(CODE (H$)=40)
 930 LET L=L+3*(CODE (J$)=39)+5*
(CODE (J$)=40)
 940 LET L=L+3*(CODE (K$)=38)-2*
(CODE (K$)=39)
 950 LET L=L+3*(CODE (L$)=40)-5*
(CODE (L$)=41)-10*(CODE (L$)=42)
 960 LET L=L-3*(CODE (N$)=38)-5*
(CODE (N$)=39)-10*(CODE (N$)=40)
-2*(CODE (N$)=41)
 965 IF L<L4 THEN LET L=L4
 970 IF P$="Y" THEN LET L=L+1
 980 IF Q$="Y" THEN LET L=L+1
 990 IF R$="Y" THEN LET L=L-1
1000 CLS
1010 FOR Z=1 TO 5
1020 PRINT
1030 NEXT Z
1040 PRINT "  [redacted]
[redacted]"
1050 PRINT
1060 PRINT "YOUR PREDICTED AGE A
T [redacted]"
1070 PRINT
1080 PRINT TAB (8);T$;", IS ";L
1090 PRINT
1100 PRINT
1110 PRINT "[redacted]
[redacted]"
1999 STOP
2000 CLS
2010 FOR Z=1 TO INT (RND*8)
2020 PRINT
2030 NEXT Z
2040 GOSUB 3000
2070 RETURN
3000 LET Z=INT (RND*10)
3005 PRINT TAB (5);
3010 GOSUB 3000+20*Z
3015 RETURN
3020 PRINT "I SEE"
3030 RETURN
```

```
3040 PRINT "UH HUH"
3050 RETURN
3060 PRINT "RIGHT"
3070 RETURN
3080 PRINT "OK"
3090 RETURN
3100 PRINT "FINE, ";T$
3110 RETURN
3120 PRINT "THANKS, ";T$;", NOW.
.."
3130 RETURN
3140 PRINT "NOT TOO MANY MORE TO
 GO"
3150 RETURN
3160 PRINT "MMM..."
3170 RETURN
3180 PRINT "THANK YOU, ";T$;", N
OW"
3190 RETURN
3200 PRINT "ALL RIGHT, NOW"
3210 RETURN
```

RUSSIAN ROULETTE

A simple little program for a 1K ZX81/new ROM ZX80, in which you must pull the trigger 10 times (by pressing NEWLINE) to survive.

```
  1 LET Q=10
  5 LET M=2
 10 FOR A=Q/Q TO Q
 20 INPUT A$
 30 CLS
 35 GOSUB 150
 40 IF RND<.16 THEN GOTO 70
 50 PRINT A,"
 60 NEXT A
 65 IF RND>=.16  THEN GOTO 90
 70 PRINT AT Q,Q/M;"
"
 75 PRINT AT Q,Q/M;"
"
 80 GOTO 70
 90 PRINT AT Q,Q/M;"YOU
"
 110 PRINT AT Q,Q/M;"
RVIVED"
 130 GOTO 90
 150 FOR T=Q/Q TO Q
 160 PRINT AT M+M,Q+M;"",CHR$ (
A+156)
```

```
170 PRINT AT M+M,Q+M;"▒","  "
175 NEXT T
180 RETURN
```

JUPITER LANDER

Forget about lunar landers. This one is set above Jupiter. Once you've managed to land your tumbling space craft successfully a number of times, make things more difficult by reducing your starting fuel (line 330) or by changing your initial speed (line 340). This needs more than 1K.

```
 10 GOSUB 320
 42 LET M=11+RND*5
 43 PRINT AT Y,Q,"    "
 45 PRINT AT 16-H/100,M;
 50 LET Q=M
 60 LET Y=16-H/100
100 LET J=RND
110 IF J<.5 THEN PRINT "▒▒"
120 IF J>.5 THEN PRINT "▒▒"
160 PRINT AT 17,0;"▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒"
165 PRINT "HEIGHT     FUEL    SPE
ED"
167 PRINT "   ";INT H;"      ";INT
F;"      ";INT S;"      "
168 PRINT AT 20,5;"THRUST?"
170 INPUT T
175 PRINT AT 20,5;T;"        "
180 IF F-T<1 THEN LET T=0
190 LET S=S+INT (S/10)+15-T
200 LET H=H-S
210 IF H>1600 THEN GOTO 290
220 LET F=F-ABS (T/2)
230 IF H>0 THEN GOTO 20
240 IF S>10 THEN GOTO 270
250 PRINT "SUCCESSFUL LANDING "
;F*23;" POINTS";W
270 PRINT "CRASH LANDING, FORMI
NG CRATER",INT (S*RND*7);" METRE
S DEEP ";W
300 PRINT ABS S;" ESCAPE VELOCI
TY ";
310 GOTO 300
320 LET H=1400+RND*100
330 LET F=90+RND*75
340 LET S=10+RND*10
345 LET Y=5
346 LET Q=S
350 RETURN
```

MINIVADERS

The minivaders march from right to left. "5" and "8" move your base right and left, and "0" fires at them. This will run, unmodified, on a new ROM ZX80, or a 1K ZX81.

```
   10 LET S=0
   20 LET P=16
   25 FOR L=1 TO 4
   30 LET A$="V  V  V  V  V  V  V
   V  V  VV  "
   40 LET F=0
   50 PRINT AT L,0;A$;AT 7,P;"$"
   60 IF F THEN GOTO 120
   70 PAUSE 20
   80 IF INKEY$="0" THEN GOSUB 23
0
   90 LET P=P+2*(INKEY$="8")-2*(I
NKEY$="5")
  100 LET A$=A$(2 TO 31)+A$(1)
  105 CLS
  106 PRINT S
  107 LET S=ABS (S-1)
  110 GOTO 50
  120 LET Y=Y-1
  140 IF Y=L THEN GOTO 170
  150 PRINT AT Y,X;"."
  160 GOTO 70
  170 IF A$(X)=" " THEN GOTO 40
  180 LET A$(X)=" "
  190 LET S=S+10
  210 IF RND>.7 THEN GOTO 270
  220 GOTO 40
  230 LET F=1
  240 LET Y=6
  250 LET X=P
  260 RETURN
  270 LET S=S+50
  280 NEXT L
```

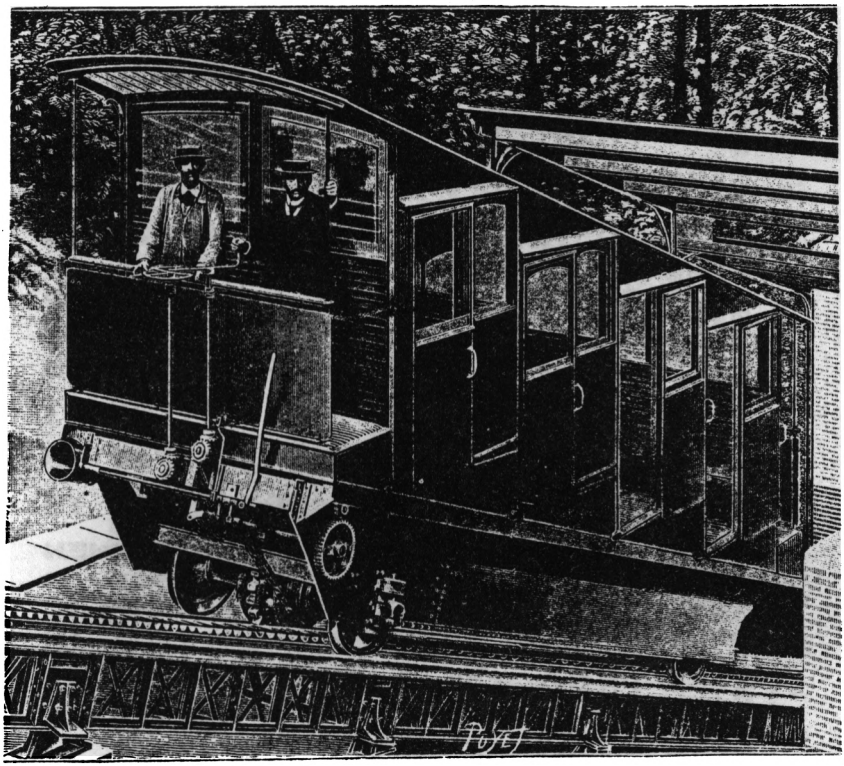
SMUGGLERS BOLD

This is a rather daft ADVENTURE-type game which demands a 16K ZX81, or ZX80 with new ROM. You can replace the PAUSE lines with FOR/NEXT loops if you want it to run more smoothly on a ZX81.

You are exploring a cave system in Penzance, with two companions of your choice, looking for treasure. An enormous number of obstacles stand in your way, but if you are brave, you will finally see the light of day.

Because some of the inverse print statements are a little difficult to read, we'll print them now, before the listing:

1100 YOU ARE IN CAVE NUMBER
1160 YOUR COMBINED MAGIC POWER
5232 ---STAND BY---
5542 GHOST OF LONG JOHN SILVER
5560 AHA, A TREASURE MAP. . .CAN YOU, READ IT ---WE SHALL SEE
5567 NO, IT HAS FADED. . .
5580 GADZOOKS . . .A SAND TROLL
5604 IT HOLDS A GENIE
5607 IT HELD NOTHING BUT STALE SMOKE
5620 . . .WILL YOU MAKE IT
5790 DESPAIR, JUST SAND
5792 AND SPIDERS
5806 SILVER, GOLD AND GEMS
7000 CONGRATULATIONS
8240 TIME TO PICK A PARTY TO EXPLORE THE CAVES WITH YOU. . .



```
  1 REM ■SMUGGLERS BOLD■
  2 REM
  3 REM REPLACE THE PAUSE LINES
  4 REM WITH LOOPS TO RUN MORE
  5 REM SMOOTHLY ON A ■ZX81■
  6 REM
  7 SLOW
 10 RAND
100 GOSUB 9500
300 REM STATE OF PARTY
305 CLS
310 IF CAVE<1 THEN LET CAVE=1
320 IF CASH<1 THEN LET CASH=0
330 IF S<1 THEN LET S=0
340 IF P<1 THEN LET P=0
350 IF CAVE>9 THEN GOTO 7000
```

```
1100 PRINT "[illegible]
[illegible]";CHR$ (156+CAVE)
1120 IF CASH>0 THEN PRINT "YOUR
PARTY IS CARRYING","£";CASH;" WO
RTH OF TREASURE"
1140 PRINT "    ";A$;" AND ";B$
1145 PRINT TAB (8);"ARE WITH YOU
"
1150 PRINT "YOUR TOTAL STRENGTH
IS ";S
1160 PRINT "YOUR COMBINED [illegible]
[illegible] IS ";P
3000 REM CAVE
3100 PRINT ,"POINTS:  ";10*CASH+2
0*S+30*P
3120 PRINT TAB (8);"▄▄▄▄▌ ▐▄▄▄▄ "
3130 IF D$="S" THEN PRINT TAB (8
);"▌   ▀   ▐ "
3150 PRINT TAB (8);"▌    ";CHR$ (
CAVE+156);"     ▐ "
3170 IF D$="E" THEN PRINT TAB (8
);"▀▀ ▀  ▀▀ "
3190 IF D$="W" THEN PRINT TAB (8
);"▀    ▀▀  "
3200 IF D$<>"W" AND D$<>"E" THEN
 PRINT TAB (8);"▀▀ "
3210 PRINT TAB (8);"▛         ▜ "
3220 PRINT TAB (8);"▌         ▐ "
3230 IF D$="N" THEN PRINT TAB (8
);"▌   ▀   ▐ "
3240 PRINT TAB (8);"▀▀▀▀▘ ▝▀▀▀▀ "
5100 PRINT "  WHICH EXIT (N,S,E,
W)?"
5120 INPUT D$
5125 IF D$="W" THEN STOP
5200 REM QUESTIONS
5210 CLS
5220 GOSUB 8000
5221 IF D<3 THEN GOSUB 5513
5225 IF D>14 THEN GOSUB 5780
5230 IF D>2 AND D<15 THEN GOSUB
5480+20*D
5232 PRINT ,,,"[illegible]"
5235 PAUSE 300
5240 GOTO 300
5513 PRINT "AHEAD OF YOU IS A CH
EST"
5514 GOSUB 8000
5515 PRINT "IT CONTAINS ";D/2;"
KG OF GEMS"
5516 PRINT "YOUR AVAILABLE STREN
GTH IS ";S
5517 PRINT "(THE MOST YOU CAN TA
KE IS ";INT (S/4);")"
5520 PRINT "HOW MUCH WILL YOU TA
KE?"
5521 INPUT DD
5522 IF DD>D/2 OR DD>INT (S/4) T
HEN GOTO 5521
5523 LET S=S-DD
5524 LET CASH=CASH+2.5*DD
```

```
5539 RETURN
5540 PRINT "YOU ARE FACE TO FACE
 WITH THE"
5542 PRINT "  GHOST OF [illegible]
 [illegible]..."
5544 PRINT "WILL HE HARM YOU?"
5545 PAUSE 100
5547 GOSUB 8000
5549 IF D>8 THEN PRINT "YES...HE
 STRIKES..."
5550 IF D>8 THEN PRINT "AND ";A$
;" IS INJURED"
5551 IF D>8 THEN LET S=S-2
5552 IF D>16 THEN LET P=P-1
5553 IF D<9 THEN PRINT "NO, HE H
AS TURNED AWAY","YOU MAY PROCEED
"
5554 IF D>12 THEN LET CAVE=CAVE-
1
5555 IF D<9 THEN LET CAVE=CAVE+1
5556 RETURN
5560 PRINT "[illegible]"..."[illegible]
[illegible]"
5562 GOSUB 8000
5564 PAUSE 100
5566 IF D<11 THEN LET S=S-INT (D
/3)
5567 IF D<11 THEN PRINT "[illegible]
[illegible]"
5568 IF D>10 THEN PRINT "YES £££
£££££ AND IT TELLS","YOU WHERE
TO FIND £";10*D/2,"WORTH OF GOLD
 DUBLOONS"
5569 IF D>10 THEN LET CASH=CASH+
10*D/2
5570 IF D>10 THEN LET CAVE=CAVE+
1
5572 RETURN
5580 PRINT "[illegible]
[illegible]","HAS IT SPOTTED YOU?"
5582 GOSUB 8000
5583 PAUSE 200
5584 IF D<7 THEN PRINT "LUCKY...
HE HAS TURNED AWAY"
5585 IF D>6 THEN PRINT "HORRORS,
 HE ATTACKS ";A$
5586 IF D>6 THEN LET CASH=CASH-1
0*D
5587 IF CASH>0 AND D>6 THEN PRIN
T "STEALING £";10*D;" WORTH OF E
MERALDS"
5589 IF D<7 THEN LET CAVE=CAVE+1
5590 LET A$="BATTERED "+A$
5595 RETURN
5600 PRINT "YOU FIND A BOTTLE FA
SHIONED OF","CURIOUS, MISTY GLAS
S...WHAT IS","INSIDE IT?"
5602 GOSUB 8000
5603 PAUSE 15*D
```

```
5604 IF D<7 THEN PRINT "IT HOLDS
 A [redacted] WHO GIVES","YOU £";5*D;
" AND INCREASES YOUR","STRENGTH
BY ";D
5605 IF D<7 THEN LET CASH=CASH+5
*D
5606 IF D<7 THEN LET S=S+D
5607 IF D>6 THEN PRINT "[redacted]
[redacted]"
5608 IF D<7 THEN LET CAVE=CAVE+1
5609 RETURN
5620 PRINT "A WILD WOLF HAS BEEN
 TRAPPED FORYEARS IN THIS CAVE.
IT IS","BREAKING FREE...YOUR PAR
TY RUNS FOR SAFETY [redacted]
[redacted]"
5622 GOSUB 8000
5623 PAUSE 250
5624 IF D<12 THEN PRINT "YES, YO
U MANAGE TO ESCAPE"
5625 IF D<12 THEN LET CAVE=CAVE+
2
5626 IF D>11 THEN PRINT "HORRORS
 IT HAS CAUGHT","AND WOUNDED ";B
$
5627 IF D>11 THEN LET B$="WOUNDE
D "+B$
5628 IF D>11 THEN LET S=INT (S/2
)
5629 RETURN
5640 PRINT "THE CAVE IS FULL OF
POISONED","GAS...FLEE FOR YOUR L
IVES"
5641 GOSUB 8000
5642 LET CAVE=CAVE-1
5643 LET S=S-INT (D/5)
5650 RETURN
5780 PRINT "IN THE CAVE ARE ";
5782 GOSUB 8000
5784 PRINT D+1;" SACKS...","WHIC
H ONE DO YOU DARE OPEN?"
5786 INPUT K
5787 PAUSE 150
5788 GOSUB 8000
5789 IF D>12 THEN GOTO 5800
5790 PRINT "[redacted]"
5792 IF D<4 THEN PRINT ,"[redacted]
[redacted]"
5793 LET S=S-2
5797 RETURN
5800 PRINT "YOU ARE LUCKY"
5805 IF D>16 THEN PRINT "DIAMOND
S AND RUBIES"
5806 IF D<17 THEN PRINT "[redacted]
[redacted]"
5810 GOSUB 8000
5812 LET CASH=CASH+7.5*D
5813 LET S=S+1
5814 LET P=P+1
5820 LET CAVE=CAVE+1
5900 RETURN
7000 PRINT "     [redacted]
■"
```

```
7010 PRINT
7020 PRINT "YOU, ";A$;" AND ";B$
7030 PRINT "MADE IT SAFELY OUT O
F THE"
7040 PRINT TAB (8);"CAVE SYSTEM"
7050 PRINT
7060 PRINT "YOU HAVE £";CASH;" [illegible]
[illegible]"
7070 PRINT
7080 PRINT
7090 PRINT "     AND SCORED ";10*
CASH+20*S+30*P;" POINTS"
7100 PRINT
7105 FOR A=1 TO 6
7110 PRINT "[illegible]
[illegible]"
7120 PAUSE 30
7130 NEXT A
7999 STOP
8000 REM RANDOM NUMBERS
8010 LET D=INT (RND*16)+1
8020 RETURN
8200 REM MAKE UP PARTY
8210 CLS
8220 GOSUB 8000
8230 LET CASH=39.5*D
8240 PRINT "[illegible]
[illegible]
[illegible]"
8250 PRINT
8255 FOR U=1 TO 2
8256 IF U=2 THEN CLS
8260 PRINT "YOU HAVE £";CASH
8270 PRINT
8280 PRINT "YOU CAN HIRE ANY [illegible]
 OF THESE:"
8290 PRINT
8300 PRINT "NAME      £ COST STRE
NGTH  POWER"
8310 PRINT
8320 PRINT "1: MORGAL   100    12
     0"
8330 PRINT "2: MERLIN    47     2
    10"
8340 PRINT "3: MUMBLE    83     6
     6"
8350 PRINT "4: HACKTO    90     9
     3"
8360 PRINT "5: MINMUK    64     3
     9"
8370 PRINT
8380 PRINT "CHOICE ";U
8385 INPUT G(U)
8390 GOSUB 8500+10*G(U)
8410 NEXT U
8412 RETURN
8415 STOP
8510 LET CASH=CASH-100
8511 LET S=S+12
8512 IF U=1 THEN LET A$="MORGAL"
8513 IF U=2 THEN LET B$="MORGAL"
8519 RETURN
```

```
8520 IF W=1 THEN LET A$="MERLIN"
8521 LET CASH=CASH-47
8522 LET S=S+2
8523 LET P=P+10
8524 IF W=2 THEN LET B$="MERLIN"
8527 RETURN
8530 IF W=1 THEN LET A$="MUMBLE"
8531 LET CASH=CASH-83
8532 LET S=S+6
8533 LET P=P+6
8534 IF W=2 THEN LET B$="MUMBLE"
8535 RETURN
8540 IF W=1 THEN LET A$="MACKTO"
8541 LET CASH=CASH-90
8542 LET S=S+9
8543 LET P=P+3
8544 IF W=2 THEN LET B$="MACKTO"
8547 RETURN
8550 IF W=1 THEN LET A$="MINMUK"
8551 LET CASH=CASH-64
8552 LET S=S+3
8553 LET P=P+9
8554 IF W=2 THEN LET B$="MINMUK"
8557 RETURN
9490 STOP
9500 REM VARIABLES
9510 DIM G(2)
9540 LET D$=""
9560 LET P=0
9570 LET S=0
9610 LET CASH=0
9620 LET CAVE=1
9890 GOSUB 8200
9900 RETURN
```

SIMON

In this 1K ZX81 program, you have to copy the sequence of numbers selected by the computer. The numbers are from one to four, and -- as you'll see when you run the program --they are printed in a position relating to the number, which makes it easier to remember the sequence.

When you run it, a single number will appear, then be blacked out. Just touch the same number on the keyboard. The same number will appear again, be blacked out, and a second number appear. You have to touch both numbers,

in order, to continue playing. And so on. If you remember seven in a row, you win. Otherwise, the program stops, showing you your score. Some interesting space-saving techniques are used in this game.

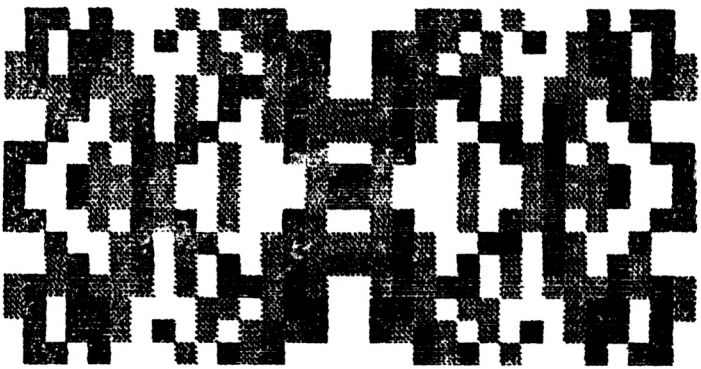
```
   5 LET A$=""
  10 LET M=7
  20 LET Z=M/M
  30 FOR A=Z TO M
  40 LET A$=A$+STR$ (INT (RND*4)
+Z)
  50 NEXT A
  60 LET X=Z
  70 FOR Q=Z TO X
  75 LET L=4*(CODE A$(Q)-29)
  80 PRINT AT L,M;A$(Q)
  90 FOR J=Z TO 20-X
 100 NEXT J
 102 PRINT AT L,M;"■"
 103 LET K=RND*RND
 105 CLS
 110 NEXT Q
 120 FOR B=Z TO X
 122 IF INKEY$<>"" THEN GOTO 122
 124 IF INKEY$="" THEN GOTO 124
 125 CLS
 130 PRINT AT 4*(CODE INKEY$-29)
,M;INKEY$
 140 IF CODE INKEY$<>CODE (A$(B)
) THEN GOTO 300
 150 NEXT B
 155 IF X=M THEN PRINT "[illegible]"
;C
 160 LET X=X+Z
 162 CLS
 165 FOR W=Z TO M+M
 166 NEXT W
 170 GOTO 70
 300 PRINT "YOU SCORED ";X-Z
```

GRAND MASTER

Here are five programs to produce pattern demonstrations on the ZX81. All except SNOWFLAKE fit within 1K.

PERPETUA

This program selects graphics or spaces from a string (A$, assigned in line 10) and PRINT ATs this in a balanced, and quite pleasing manner.



```
 10 LET A$="
"
 20 LET B$=A$(INT (RND*23)+1)
 30 LET A=RND*15
 40 LET B=RND*31
 50 LET C=15
 60 LET D=31
 70 PRINT AT A,B;B$
 80 PRINT AT C-A,B;B$
 90 PRINT AT C-A,D-B;B$
100 PRINT AT A,D-B;B$
110 RUN
```

SNOWFLAKE

This uses PLOT and UNPLOT to create a balanced pattern

inside a frame . It needs more than 1K.

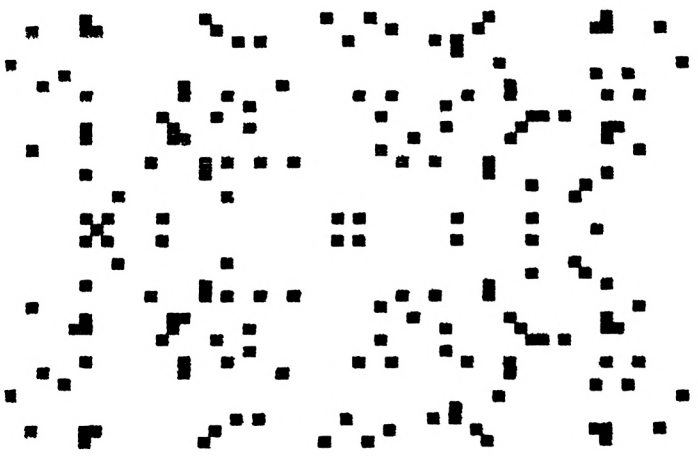
```
  1 GOSUB 500
  5 RAND
 10 LET A=60*RND+1
 20 LET B=40*RND+1
 30 IF RND>.5 THEN GOTO 120
 50 PLOT A,B
 65 PLOT A,42-B
 70 PLOT 64-A,B
 90 PLOT 64-A,42-B
110 GOTO 10
120 UNPLOT A,B
140 UNPLOT A,42-B
160 UNPLOT 64-A,B
180 UNPLOT 64-A,42-B
200 GOTO 10
500 FOR J=1 TO 42
520 PLOT 2,J
525 PLOT 1,J
530 PLOT 62,43-J
535 PLOT 63,43-J
550 NEXT J
570 FOR J=1 TO 62
580 PLOT J,0
585 PLOT J,1
590 PLOT 63-J,42
595 PLOT 63-J,41
610 NEXT J
615 PLOT 63,0
620 RETURN
```

PEA POD

This uses PLOT, but is designed not to produce a balanced pattern, as the sample run indicates. The LET K = RND*RND lines are included to slow things down. By all means leave them out if you're impatient.

```
 31 LET C=62
 32 LET D=40
 35 LET A=C*RND
 40 LET B=D*RND
 50 LET K=RND**RND
 60 PLOT A,B
 70 LET K=RND**RND
 80 PLOT A,D-B
 90 IF RND>RND THEN RUN
100 PLOT C-A,B
```

```
110 LET K=RND**RND
120 PLOT C-A,D-B
130 LET K=RND**RND
135 RUN
```



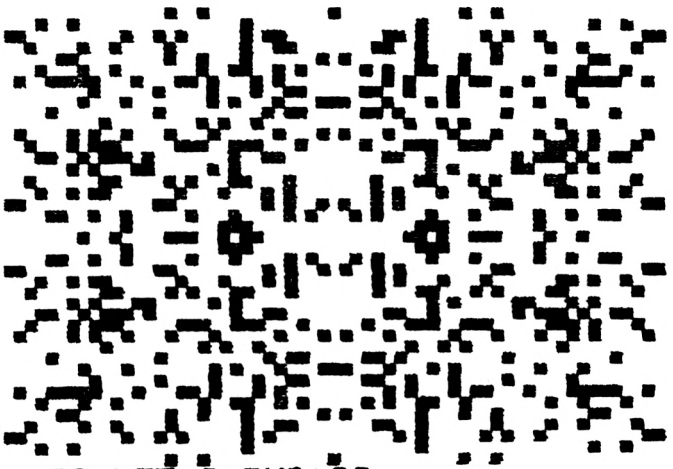
SCARSDALE

Scarsdale uses PRINT AT, but prints in pairs, that is, locations above and below each other (as will be clear when you run it). B$, in line 20, is an inverse space, a space, and a graphics A.

```
 20 LET B$="▒ ▒"(INT (RND*3)+1)
 30 LET A=RND*13
 40 LET B=RND*30
 50 LET C=14
 60 LET D=31
 70 PRINT AT A,B;B$
 75 PRINT AT A+A/A,B;B$
 80 PRINT AT C-A,B;B$
 85 PRINT AT C-A+A/A,B;B$
 90 PRINT AT C-A,D-B;B$
 95 PRINT AT C-A+A/A,D-B;B$
100 PRINT AT A,D-B;B$
105 PRINT AT A+A/A,D-B;B$
110 RUN
```

BLIP BLIP

Blip blip, which is based on the BUTTERFLY program in Tim Hartnell's book GETTING ACQUAINTED WITH YOUR ZX81, produces the pattern shown below, perfectly balanced, but with no UNPLOT facility, so eventually the screen will turn completely black.



```
 30 LET A=RND*60
 40 LET B=RND*40
 50 LET C=60
 60 LET D=40
 70 PLOT A,B
 80 PLOT C-A,B
 90 PLOT C-A,D-B
100 PLOT A,D-B
110 RUN
```

WARPO

The WARPO is a spooky-looking alien who pops up at random on the numbers one to five. If you press the same number before he vanishes, he will turn into a checkerboard pattern, and your score will increase. He will appear 20 times

in a round. A good score is over 190. This takes 1K on a new ROM ZX80 or ZX81 and must be run in the FAST mode.

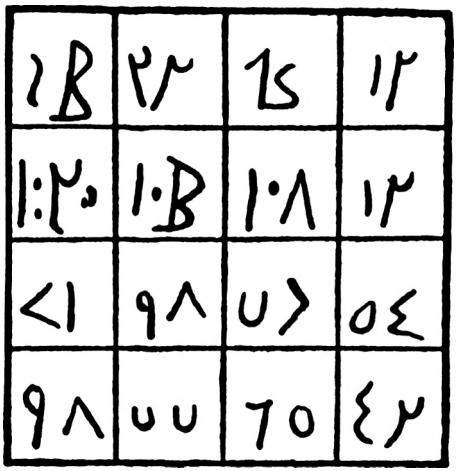
```
 10 DIM A$(2,2,5)
 20 LET A$(1,1)="     "
 30 LET A$(1,2)="     "
 40 LET A$(2,1)="     "
 50 LET A$(2,2)="     "
 60 PRINT AT 4,0;"   1     2     3
    4     5"
 70 LET S=0
 80 FOR A=0 TO 19
 90 PRINT AT 5,0;S
100 LET T=INT (RND*2)+1
110 LET P=INT (RND*5)
120 PRINT AT 0,P*5;A$(T,1)
130 PRINT TAB P*5;A$(T,2)
140 PAUSE 120-A*5
150 IF INKEY$<>STR$ (P+1) THEN
GOTO 200
160 LET S=S+T*10
170 PRINT AT 0,P*5;"     "
180 PRINT TAB P*5;"     "
190 PAUSE 30
200 PRINT AT 0,P*5;"     "
210 PRINT TAB P*5;"     "
220 NEXT A
```

BOWLING

This program, which allows you to emulate (after a fashion) a 10 pin bowling alley, is written for a ZX81 with more than 1K of memory. If you convert it for the ZX80, it will fit into 1K. You bowl by hitting NEWLINE. There are 10 frames in a game, two balls in a frame. You get a bonus of 15 if you get all 10 pins down with two balls, and a bonus of 30 if you get the pins down with the first ball. There is a highest score feature.

```
10      DIM A(10)
20      LET Y = Ø
30      LET S = Ø
40      FOR B = 1 TO 10
50      FOR E = 1 TO 2
60      PRINT AT Ø,Ø;
70      PRINT "FRAME ";B,"BALL ";CHR$(E + 156)
80      LET Z = 0
90      FOR C = 1 TO 10
```

```
100     IF E = 2 THEN GOTO 120
110     LET A(C) = 52
120     IF A(C) = 52 AND RND > .5 THEN LET A(C) = 61
130     IF A(C) = 61 THEN LET Z = Z + 1
140     NEXT C
150     PRINT CHR$(A(10));" * ";CHR$(A(9));" * ";CHR$(A(8));
        " * ";CHR$(A(7))
160     PRINT " * "; CHR$(A(6));" * ";CHR$(A(5));" * ";
        CHR$(A(4))
170     PRINT " * * ";CHR$(A(3));" * ";CHR$(A(2))
180     PRINT "   * * * ";CHR$(A(1))
190     PRINT
200     PRINT
210     PRINT "SCORE THIS FRAME ";Z
220     IF E = 1 AND Z = 10 THEN GOTO 370
230     IF Z > 9 THEN LET Z = 15
240     IF E = 2 THEN LET S = S + Z
250     PRINT "SCORE SO FAR ";S
260     INPUT U$
280     NEXT E
290     NEXT B
300     PRINT "SCORE FOR THAT GAME WAS ";S
310     IF S < Y THEN GOTO 330
320     LET Y = S
330     PRINT "HIGHEST SCORE SO FAR ";Y
340     INPUT U$
350     CLS
360     GOTO 30
370     CLS
380     PRINT "STRIKE"
390     FOR R = 1 TO 200
400     NEXT R
410     LET S = S + 15
420     LET E = 2
430     CLS
440     GOTO 230
```



SPEEDWAY

Here are three programs in which you have to drive a car down a track which twists and turns. Each fit within 1K on a ZX81, and use different tricks to squeeze within the limited memory.

RACER

In this, your car is an inverse H (see line 110). The graphics in line 60 are inverse space, three graphic H, inverse space; and the graphic in line 70 is from the H key. Use the graphic A if you want to leave a trail. The spot in front of your car turns into your score. Any total over 236 is very good. The "M" and "Z" keys will control your vehicle.

```
  20 LET A=10
  22 LET Z=A/A
  25 LET U=A/2
  30 LET B=A
  40 LET C=A+A
  45 LET T=Z
  50 LET D=A-A/A
  60 PRINT AT C,D;"▒▒▒▒▒▒▒▒▒"
  70 PRINT AT A,B;"▒"
  80 SCROLL
  90 IF INKEY$="Z" THEN LET B=B-
Z
 100 IF INKEY$="M" THEN LET B=B+
Z
 110 PRINT AT A,B;"▒"
 120 IF D<A+U THEN LET D=D+2*RND
 125 IF D>U THEN LET D=D-2*RND
 140 PRINT AT A+Z,B;
 150 IF PEEK (PEEK 16398+PEEK 16
399*256)=128 THEN PRINT T;W
 155 LET T=T+Z
 160 GOTO 60
```

BRANDS HATCH

Your car is a V this time, and the track is clear. Again "Z" and "M" control your vehicle, but the INKEY$ is interpreted in a different way from that in RACER (compare line 90 of BRANDS HATCH with lines 90 and 100 of RACER).

```
  20 LET A=10
  25 LET Z=A/A
  30 LET B=A
  35 LET Y=B-B
```

```
  40 LET C=A+A
  50 LET D=A
  60 PRINT AT C,D;"■■    ■■"
  70 PRINT AT A,B;"  "
  80 SCROLL
  90 LET B=B-(INKEY$="Z")+(INKEY
$="M")
 110 PRINT AT A,B;"U"
 120 IF D<17 THEN LET D=D+2*RND
 125 IF D>7 THEN LET D=D-2*RND
 140 PRINT AT 11,B;
 150 IF PEEK (PEEK 16398+PEEK 16
399*256)=128 THEN PRINT Y;U
 155 LET Y=Y+Z
 160 GOTO 60
```

WISE-MAN

The vehicle (a Y, which explains the title) is not "unPRINTed" so the Y turns into a long, swinging line of Y's which you must guide through the terrors of a long and winding road (two graphic A's, separated by two spaces).

The "5" and "8" keys control the Y's. Line 40 changes the line from which SCROLL operates. This technique was discovered by Alastair Gourlay (author of 30 AMAZING GAMES FOR THE 1K ZX81). This program fits a 1K ZX81.

```
  10 LET A=5
  20 LET J=A/A
  30 LET S=A+A
  40 POKE 16418,A
  50 LET K=S+PI
  70 LET D=K
  80 LET P=J
  90 SCROLL
 100 PRINT TAB D;"■  ■"
 110 LET D=D+RND*2*(NOT D>18)-RN
D*2*(NOT D<6)
 130 PRINT AT S,K;"Y"
 140 PRINT AT S+J,K;
 150 IF PEEK (PEEK 16398+256*PEE
K 16399)=136 THEN GOTO 200
 170 LET P=P+1
 180 LET K=K+(INKEY$="8")-(INKEY
$="5")
 190 GOTO 90
 200 PRINT P
```

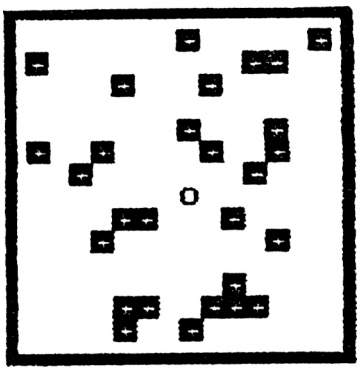
SNAP

The 1K ZX80 (new ROM) or ZX81 generates a number of large print numbers (one to nine) in fairly rapid succession. You have to try and anticipate which number will be next by holding it down. If the computer's next number is the same as the one you are holding, the word SNAP is printed in large letters (using the routine starting at 33). There are 10 numbers to a round. At the end of the round, your score is printed as a giant number. On a ZX81, you must run this in FAST.

```
 1 DIM A$(4,2)
 2 DIM B$(50)
 3 LET A$(1)="▛▀"
 4 LET A$(2)="▌ "
 5 LET A$(3)=" ▐"
 6 LET A$(4)="▌▐"
 7 LET B$="1444133333131211313
144133121311214113333141411413 1"
 8 GOTO 20
10 FOR I=1 TO 5
11 PRINT A$(VAL B$(5*X+I))
12 NEXT I
13 RETURN
20 LET S=0
21 FOR G=1 TO 10
22 LET X=INT (RND*10)
23 GOSUB 10
24 PAUSE 50
25 IF INKEY$=STR$ X THEN GOSUB
 33
26 CLS
27 NEXT G
28 CLS
29 PRINT "YOU SCORED"
30 LET X=S
31 GOSUB 10
32 STOP
33 PRINT AT 1,16;"▗▄▄ ▙ ▌▗▄▄▖
▛▀▙"
34 PRINT TAB 16;"▀▀▙ ▌▀▙▌▙▄▄▌▙
▄▘"
35 PRINT TAB 16;"▚▄▄▘ ▌ ▜▌ ▌▌
"
36 LET S=S+1
37 PAUSE 50
38 RETURN
```

HADYN

This game -- as you can see from the screen printout -- puts you (the "O") within a frame, with inverse asterisks appearing



SCORE 32

every second or so. If one lands on you, you are dead, and the game is over. The "W" key moves you up, "D" to the right, "A" to the left and "X" moves you down. You should be able to accumulate a score of more than 115.

```
  10 PRINT "┌──────────────────────┐"
  20 FOR N=1 TO 14
  30 PRINT "│                      │"
  40 NEXT N
  50 PRINT "└──────────────────────┘"
  60 LET P=16916
  70 LET S=0
  80 POKE P,52
  90 PRINT AT 20,0;"SCORE ";S
 100 PRINT AT 1+(RND*13),1+(RND*
13);"■"
 110 IF PEEK P<>52 THEN STOP
 120 PAUSE 30
 130 POKE P,0
 140 LET S=S+1
 150 LET P=P+(INKEY$="D")-(INKEY
$="A")+17*((INKEY$="X")-(INKEY$=
"W"))
 160 IF PEEK P<>0 THEN STOP
 170 GOTO 80
```

UFO

You are in command of a UFO in this 1K game, and little aliens (the letter As) are running about underneath. You have a limited time in which to destroy them. You fire by pressing "1" but each shot drastically reduces the length of time you have left. "5" and "8" move you back and forth. Enter a number between two and 31 at the start of the game. This number dictates the speed and direction of the aliens' movement. Your score is shown at the top of the screen.

```
  10 LET A=0
  15 INPUT W
  20 LET B$="A     A       A       A
   A     AA      "
  30 LET Z=0
  50 FOR J=1 TO 1000
 120 FOR E=12 TO 18 STEP 2
 125 PRINT AT 10,A;"      "
 127 LET Q=ABS (A)+2
 130 IF INKEY$<>"1" THEN GOTO 14
2
 132 LET J=J+10
 135 PRINT AT E,Q;"■"
 137 IF B$(Q-1)="A" THEN LET Z=Z
+3469
 140 PRINT AT E,Q;" "
 142 NEXT E
 144 IF INKEY$="1" THEN LET B$(Q
-1)="*"
 150 PRINT AT 0,9;Z
 160 PRINT AT 20,0;B$
 170 LET B$=B$(W TO )+B$(1 TO W-
1)
 490 LET A=A-(INKEY$="8")+(INKEY
$="5")
 500 NEXT J
```

SQUASH

There are three balls to a round. The "7" moves your bat up, "6" moves it down. The longer you can keep the ball in play,

the better your score. Any score over 67 is good. This takes 1K on a ZX81/new ROM ZX80. Run it in FAST.

```
  1 LET T=0
 10 LET S=3
 20 LET B=11
 30 LET C=B
 40 LET A=12
 45 PRINT "LIVES ";S
 50 PRINT AT B,13;" "
 60 PRINT AT C,A;" "
 65 IF A=0 THEN LET C=INT (RND*
5)+10
 70 LET A=A-1
 80 IF INKEY$="7" THEN LET B=B-
1
 90 IF INKEY$="6" THEN LET B=B+
1
100 PRINT AT B,13;"■"
105 LET A=A-1
110 PRINT AT C,A;"O"
120 IF A=-12 AND B=C THEN LET A
=12
125 LET T=T+1
130 IF A=-20 THEN GOTO 200
150 GOTO 50
200 LET S=S-1
205 CLS
210 IF S>0 THEN GOTO 20
240 PRINT "YOU SCORED ";T
```

CONVERTING PROGRAMS FOR THE NEW ROM AND THE ZX81

It is generally fairly easy to convert programs written for old ROM ZX80's so they will run on the new ROM machines, although programs which include PEEKing and POKEing — especially into REM statements — can cause some difficulties.

In many cases, you'll find the program runs far better on a new ROM machine. That is, it is more effective in giving user prompts, or a moving display, or in screen layout.

However, the new ROM is not as generous in memory as was the old ROM. Many 1K programs will not fit into 1K on the new ROM, because the systems variables for the new ROM use up much more of the original 1K than they do on the 4K ROM machine.

MOVING DISPLAY

Omit a moving display routine completely if inputting on old ROM program to a new ROM machine. Change the 'time' figure (usually given in the GOSUB line POKE 16414, n (where n is the 'time' figure). In other programs, the time is defined as LET T = n. Using the PAUSE functions, set n (as in PAUSE n) to give the most effective display, bearing in mind that PAUSE 50 (PAUSE 60 in the U.S.) will hold the display for one second, pause 25 (PAUSE 30) for half a second, PAUSE 100 (PAUSE 120) for two seconds and so on. You can get a guide as to which figure to place after the word PAUSE by keeping in mind that the higher the number used for time in the old ROM moving display (up to, and including, 254), the shorter the time the display is held.

RANDOM NUMBERS

It is more expensive — in memory terms — to generate a random number on the new ROM than it is on the old.

Here are the two versions:

OLD ROM	*NEW ROM*
LET J = RND(6)	*LET J = INT(RND*6) + 1*

Some memory is saved by the fact that INT and RND are

just single keystroke entries on new ROM machines, so each occupies just one byte. However, the multiplication sign, and the addition (which stops you getting Ø as part of your sequence of random numbers) eat up memory. If you need to generate many random numbers within different ranges during the course of a program, it might be worth having a subroutine of the type: LET J = INT(RND*K) + 1, and assign K each time before GOSUBing. Of course, this will take longer than having the line in sequence, but if operating time is not vital (and it rarely is unless you're using an animated display), you'll find it will save you programming time and — in some programs — will also save you space if you do more in the subroutine than just generate random numbers.

```
10  PRINT RND;" * "; (where * is a single space)
20  PAUSE 40
30  GOTO 10
```

This sequences can by very useful. For example, the line on the old ROM; IF RND(2) = 1 THEN... can easily be changed to the (almost) identical IF RND > .5 THEN... You can also use this to make statistically weighted decisions, so that if you want a particular program branch followed, say, about one in three times, you can just say: IF RND < .34 THEN .....

PRINT AT

There is a neat little routine, which used the line POKE Y*33 + X + 1 + PEEK(16396) + PEEK (16397)*256, n to POKE character n onto the screen of an old ROM ZX80 at location Y, X (Y being lines down from the top of the screen, X being spaces across from the left hand side of the display). The new ROM machines do this automatically, and also has the

advantage of allowing you to have more than one character at this location. I'll explain:

On the new ROM, you can use the PRINT AT command in the following way. You need to give two co-ordinates, Y (counted down from the top of the screen) and X (counted across from the left hand side of the screen). These are used in a line as follows, which will print the word END at about the middle of the screen:

```
10  PRINT AT 10, 14;"END"
```

You separate the two co-ordinates with a comma, and place a semi-colon after the second co-ordinate, before the words you want printed. The two co-ordinates can be worked out during the course of a program, so the line: PRINT AT B, A/3;"END" is acceptable. the function PRINT AT automatically "INTs" a non-integer co-ordinate.

So, if you need to print at a certain location, or you feel a program would be enhanced by a PRINT AT (and it can save a lot of blank PRINT lines, and loops which print single spaces across the screen), by all means do so. If you want an object to move, you'll have to add a PAUSE n command after the PRINT AT, at the same locations, with blanks occupying the same locations as those previously occupied by, for example, the work END. Here's a simple example:

```
10  LET A = 0
20  LET B = 0
30  PRINT AT A,B;"X"
40  PAUSE 30
50  PRINT AT A, B;" " (a single space between quotes)
60  LET A = A + RND
70  LET B = B + RND
80  IF A > 18 THEN LET A = 0
```

```
 90  IF B > 18 THEN LET B = 0
100  GOTO 30
```

This will make an X move erratically (more or less diagonally) across the scree. So, you can omit the long POKE Y*33...etc and replace it, on new ROM machines, with a simple PRINT AT.

The new ROM also has a TAB function which can start a PRINT statement at any point you choose on a line, without you having to use a 'blank spaces' loop.

OLD ROM

```
10  FOR A = 1 TO 10
20  PRINT "*";
30  NEXT A
40  PRINT "END"
```

NEW ROM

```
10  PRINT TAB 10;"END"
```

Note that you need a semi-colon after the number before the characters to be printed. TAB is a single entry function.

GRAPHICS

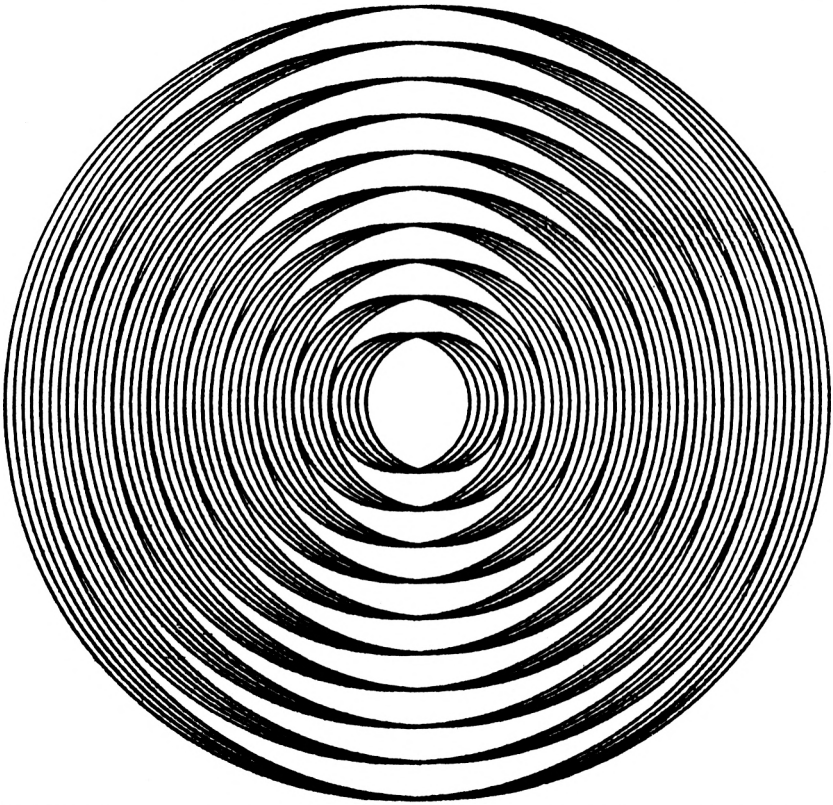
All the graphics symbols (including inverse graphics, inverse numbers and letters, even an inverse space) are available directly from the keyboard on new ROM machines. This saves use of the CHRS$(n) idea (although if you do need it, you'll be pleased to know that CHR$ is a single entry function). The 'automatic' inverse graphics can be used to 'dress up' programs by giving, for example, the instructions and user prompts in inverse letters.

To convert others, use the following table, in which the old ROM position is followed by the new ROM:

Shift Q, graphic 5; Shift W, graphic 6; shift E, graphic 1; shift R, graphic 2; shift T, graphic D; shift A, graphic A; shift S, graphic T; shift D, graphic 4; shift F, graphic 3; shift G, graphic S.

REM statements, TL$

The first address after the word REM on the old ROM is



16427. The equivalent address on the new ROM machines is 16514. You'll have to do some careful figuring to convert programs which rely heavily on data stored in a REM statement to get them to work on a new ROM machine. The TL$ (truncate left) function on the old ROM is not available on the new ROM, so user responses which are two or more

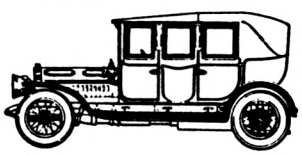
letters, which the computer assesses by stripping the string input character by character, will need to be replaced by prompts which allow the user to input (and the computer to deal with) the information single character by single character. The string arrays on the new ROM (which are actually character arrays) can be used as a form of READ/DATA. TL$(A$) can be simulated on the new ROM by A$(2 TO).

INT

As a general rule, always add the function INT (a single keystroke on the new ROM) before a division. That is, if the old ROM program says LET F = A/16, the new ROM version should be: LET F = INT(A/16). This is not needed if you're simply using the result of a calculation to PRINT AT or to TAB as the INT function is carried out automatically.

This is a conversion table for PEEK/POKE addresses used with REM statements.

OLD ROM	NEW ROM/ZX81		
16426	16513	16446	16533
16427	16514	16447	16534
16428	16515	16448	16535
16429	16516	16449	16536
16430	16517	16450	16537
16431	16518	16451	16538
16432	16519	16452	16539
16433	16520	16453	16540
16434	16521	16454	16541
16435	16522	16455	16542
16436	16523	16456	16543
16437	16524	16457	16544
16438	16525	16458	16545
16439	16526	16459	16546
16440	16527	16460	16547
16441	16528		
16442	16529		
16443	16530		
16444	16531		
16445	16532		



CODE	MEANING
Ø	Successful completion
1	NEXT with no FOR
2	Variable name not found
3	Subscript out of range, or error regarding subscript
4	Not enough room in memory
5	No more room on screen
6	Arithmetic overflow
7	RETURN with no GOSUB
8	Attempt to use INPUT in the direct mode
9	STOP statement executed
A	Invalid argument to certain functions
B	Integer out of range
C	The text of the (string) argument of VAL does not form a valid numerical expression
D	Program interrupted by BREAK or the INPUT line starts with STOP
F	Program name provided (for SAVE) is the empty string

CONT/CONTINUE is the same as GOTO m, where m is the line number displayed after the error code, except after code 9 when it is GOTO m + 1

RIP-ROARING GAMES

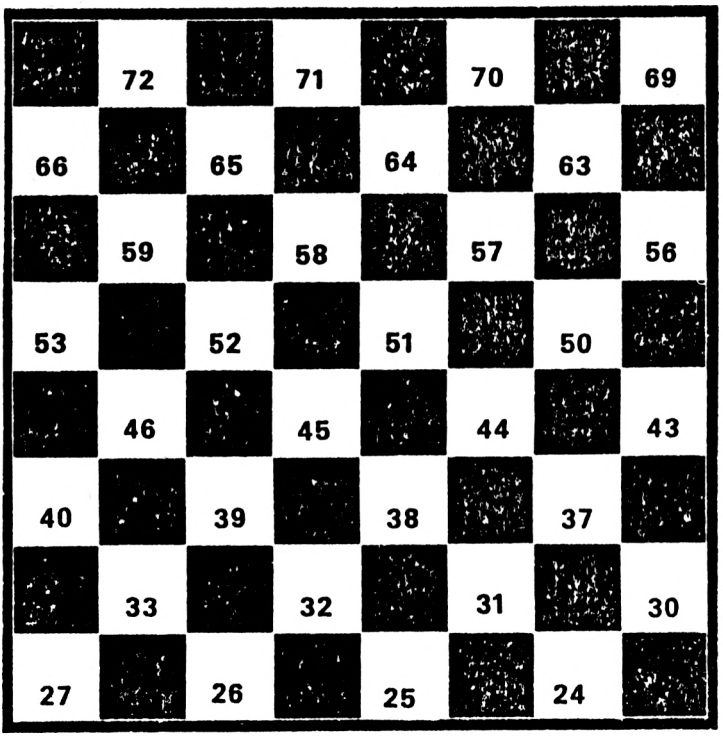
FOR THE

ZX80

GAMES

DRAUGHTS IN 1K

DRAUGHTS in 1K? We couldn't believe it could be done. But once we'd RUN this program, and discovered that it actually did play draughts/checkers, and really did take up less than 1K on the old ROM ZX80, we realised it would have to be in the book. The program is in two parts. The first sets



up the board, and the second actually plays the game. All you do is input the first program, which ends at line 13Ø (PRINT "READY") and RUN it. Once you've done this, you input the second program, which erases lines from the first program. However, the information from the first program is held in the ZX80's variables store, and — so long as you do not press RUN or CLEAR — will stay there. Then, to play

the game, you use the printed board, and buttons or coins as pieces. Your pieces start at the bottom on the page (on the low numbers) and the ZX80 starts at the top of the page. Once you've got the pieces in place, you either input GOTO 10 (if you want the computer to move first) or GOTO 320 if you insist on having first move.

The ZX80's moves are shown as two numbers. The first is the square it is moving from, and the second — naturally enough — is the square it is moving to. Move the piece on the board as instructed, and then decide on your move. Make sure you move your piece BEFORE you input your move (which you do by inputting the number of the square you're moving from, then NEWLINE, then the square you're moving to) or you may forget what your move was. At times, the ZX80 makes a decision so quickly you hardly have time to move your finger from NEWLINE before it has made, and printed, its decision.

The ZX80 will make kings automatically (and use them most effectively) but there is no provision within the program for multiple jumps by either the computer or the player. Make sure you SAVE the program before you RUN it, or you'll lose the contents of the board. You need to reLOAD this program each time before playing.

Here is the first part of the program, the section which sets up the board. If you have more than 1K, add 500 to each of the line numbers in this section, add the line 5 GOSUB 510 and change line 130 (which will be 630) into RETURN. This will mean you do not need to reLOAD each time, and you'll be able to use RUN, instead of GOTO. The ZX80 will have first move every time in the upper memory version. (Note the underlined asterisk <u>*</u> in line 110. This symbol is used throughout the book to indicate a single space

```
1Ø        DIM A(82)
2Ø        DIM X(2)
3Ø        LET X(1) = - 6
4Ø        LET X(2) = - 7
5Ø        FOR Z = Ø TO 82
6Ø        LET A(Z) = 9
7Ø        IF Z < 73 AND Z > 55 AND NOT (Z = 67 OR Z = 68
               OR Z = 6Ø OR Z = 61 OR Z = 62) THEN LET A(Z) = 1
8Ø        IF Z < 54 AND Z > 42 AND NOT (Z = 47 OR Z = 48 OR
               Z = 49) THEN LET A(Z) = Ø
9Ø        IF Z < 41 AND Z > 23 AND NOT (Z = 34 OR Z = 35 OR
               Z = 36 OR Z = 28 OR Z = 29) THEN LET A(Z) = -1
1ØØ       NEXT Z
11Ø       LET A$ = "MY MOVE * "
12Ø       LET B$ = "YOURS?"
13Ø       PRINT "READY"
```

Input the program and RUN it. SAVE this a couple of times, and then input the following program. DO NOT PRESS RUN AGAIN or you'll lose the board.

```
1Ø        LET Q = Ø
2Ø        FOR Z = 24 TO 72
3Ø        IF NOT (A(Z) = 1 OR A(Z) = 2) THEN GOTO 1ØØ
4Ø        IF A(Z) = 1 AND Z > 23 AND Z <28 THEN LET A(Z) = 2
5Ø        FOR X = 1 TO 2
6Ø        IF A(Z + X(X)) < Ø AND A(Z + 2* X(X) ) = Ø
               THEN LET Q = X(X)
65        IF Z > 55 THEN GOTO 8Ø
7Ø        IF A(Z) = 2 AND A(Z - X(X))< Ø AND A(Z - 2* X(X) )
               = Ø THEN LET Q = - X(X)
8Ø        IF NOT Q = Ø THEN GOTO 12Ø
9Ø        NEXT X
1ØØ       NEXT Z
11Ø       IF Q = Ø THEN GOTO 16Ø
12Ø       LET A(Z + Q) = Ø
13Ø       LET A(Z + 2*Q) = A(Z)
14Ø       LET A(Z) = Ø
15Ø       PRINT A$;Z, Z + 2*Q
155       GOTO 32Ø
16Ø       LET Y = Ø
17Ø       LET Z = 23 + RND(49)
18Ø       LET Y = Y + 1
19Ø       IF Y < 1ØØ AND NOT (A(Z) = 1 OR A(Z) = 2)
               THEN GOTO 17Ø
2ØØ       FOR X = 1 TO 2
21Ø       IF A(Z + X(X)) = Ø THEN LET Q = X(X)
22Ø       IF A(Z) = 2 AND A(Z - X(X)) = Ø THEN LET Q = -X(X)
23Ø       IF NOT Q = Ø THEN GOTO 29Ø
25Ø       NEXT X
26Ø       IF Y < 1ØØ THEN GOTO 17Ø
```

```
270       PRINT "YOU WIN"
280       STOP
290       LET A(Z + Q) = A(Z)
300       LET A(Z) = 0
310       PRINT A$;Z, Z + Q
320       PRINT ,B$
325       INPUT A
330       INPUT B
335       CLS
340       LET A(B) = - 1
350       LET A(A) = 0
360       IF ABS(A - B) > 7 THEN LET A(A + ((B - A)/2)) = 0
370       GOTO 10
```

Space-Station

The lives of a hundred or so colonists are in your hands. You are in control of a 4K space station, with limited food and oxygen, and subject to occasional attacks from space pirates. You make money, to buy oxygen and food, and to pay the annual maintenance bill for the station, by making and trading 'ARTEFACTS'. Now, artefacts use up oxygen when they're being manufactured, so you have to choose carefully how many to make each year. This game is a space age version of KINGDOMS and — apart from the infrequent attacks from outer space — does *not* depend on random factors during the course of a game. It is almost completely a game of skill. Unless you are very clever, or the starting parameters are particularly generous, you are unlikely to be able to keep the station alive for more than 12 years.

```
1     RANDOMISE
10    GOSUB 3000
20    LET YEAR = YEAR + 1
30    LET FOLK = FOLK + FOLK/(2 + RND(18)) - FOLK/(3 + RND(15))
40    GOTO 710
50    PRINT "COMPUTERS REPORT:"
60    PRINT
70    IF OXY < OXYNEED * FOLK THEN GOTO 8000
80    IF FOOD < FOODNEED * FOLK THEN GOTO 8100
90    IF CASH < 1 THEN GOTO 8200
100   IF FOLK < 2 THEN GOTO 8300
110   IF FOLK < 13 THEN PRINT "WARNING - POPULATION IS",
          "NEARING EXTINCTION"
120   IF OXY < 2 * OXYNEED * FOLK THEN PRINT "WARNING -
          OXYGEN SUPPLIES LOW"
```

```
130   IF FOOD< 2 * FOODNEED * FOLK THEN PRINT "WARNING -
          FOOD STOCKS LOW"
140   IF CASH <2000 THEN PRINT "WARNING - MONEY
          RUNNING LOW"
150   PRINT "**THERE ARE * ";FOLK;" * PEOPLE ON"
160   PRINT "THE SPACE STATION IN YEAR * ";YEAR;"***"
170   PRINT
180   PRINT "MONEY CREDIT IS £";CASH
190   PRINT ,"ANNUAL MAINTENANCE:  £";REPAIR
200   PRINT "OXYGEN TANKS HOLD * ";OXY;" UNITS"
210   PRINT "OXYGEN COSTS £";OXYCOST;" * PER UNIT"
220   PRINT "OXYGEN NEED PER PERSON: * ";OXYNEED
230   PRINT
240   PRINT "FOOD STOCKS STAND AT * ";FOOD

610   LET U = 128 + RND(11)
620   FOR J = 1 TO 32
630   PRINT CHR$ (U);
640   NEXT J
650   PRINT
700   RETURN
710   GOSUB 50
712   PRINT "ARTEFACTS - HOW MANY WILL YOU",,"MAKE
          AND TRADE?"
714   PRINT " * * THEY USE UP * ";ARTCOST;" UNITS OF"
718   PRINT "OXYGEN AND SELL FOR £";ARTPAY
720   INPUT B
730   IF B * ARTCOST < OXY THEN PRINT "NOT ENOUGH OXYGEN"
740   IF B * ARTCOST < OXY THEN GOTO 720
745   LET CASH = CASH + B * ARTPAY
750   LET OXY = OXY - B * ARTCOST
755   CLS
760   GOSUB 50

762   PRINT "FOOD COSTS £";FOODCOST;" * PER UNIT"
763   PRINT "EACH PERSON NEEDS * ";FOODNEED;" * FOOD UNITS"
764       PRINT "(£"; FOODCOST * FOODNEED;" * EACH, £";
              FOLK * FOODCOST * FOODNEED;" * FOR
              STATION"
766       PRINT "THIS WILL LAST * ";FOOD/ (FOODNEED * FOLK);
              " * YEARS AT THE","PRESENT POPULATION"
770       PRINT "HOW MANY FOOD UNITS WILL YOU BUY?"
780       INPUT C
790       IF C * FOODCOST <  CASH THEN PRINT "NOT ENOUGH MONEY"
800       IF C * FOODCOST <  CASH THEN GOTO 780
805       LET FOOD = FOOD + C * FOODCOST
810       LET CASH = CASH - C * FOODCOST
820       CLS
830       GOSUB 50

850       PRINT "HOW MUCH OXYGEN WILL YOU BUY?"
855       PRINT "(CURRENT STOCKS WILL LAST FOR * "; OXY/
              (OXYNEED * FOLK);" * YEARS AT THE
              PRESENT POPULATION)"
860       INPUT D
870       IF D * OXYCOST <CASH THEN PRINT "NOT ENOUGH MONEY"
880       IF D * OXYCOST <CASH THEN GOTO 860
```

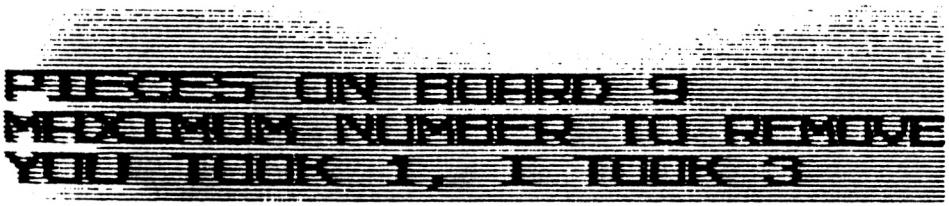
```
89Ø       CLS
9ØØ       IF RND(5) = 2 THEN GOSUB 7ØØØ

2ØØ5      LET FOOD = FOOD - FOLK * FOODNEED
2Ø3Ø      LET CASH = CASH - REPAIR - D * OXYCOST
2Ø4Ø      LET OXY = OXY + D - FOLK * OXYNEED
2Ø5Ø      GOTO 2Ø

3Ø1Ø      LET YEAR = RND(5)
3Ø15      LET A$ = "THE STATION IS DEAD"

3Ø2Ø      LET FOLK = 8Ø + RND(4Ø)
3Ø3Ø      LET CASH = 7 * (7ØØ + RND (8ØØ))/ RND(3)
3Ø4Ø      LET FOODCOST = RND(7)
3Ø5Ø      LET ARTCOST = 1 + RND(3)
3Ø55      LET FOOD = 2ØØØ + RND(5ØØ)
3Ø6Ø      LET OXY = 2ØØØ - RND(15ØØ)
3Ø7Ø      LET OXYCOST = RND(7)
3Ø8Ø      LET ARTPAY = 3Ø * RND(ARTCOST)
3Ø9Ø      LET REPAIR = 2ØØ + RND(4ØØ)
31ØØ      LET FOODNEED = 1 + RND(5)
3105      LET OXYNEED = 2 + RND(3)
3115      RETURN
7Ø1Ø       CLS
7Ø12       LET J = RND(6)
7Ø13       PRINT "THE STATION WAS ATTACKED BY"
7Ø14       IF J = 1 THEN PRINT "A FLEET OF SYRIAN SHIPS"
7Ø15       IF J = 2 THEN PRINT "RENEGADE EARTHLINGS"
7Ø16       IF J = 3 THEN PRINT "MARTIAN SPACE PILOTS"
7Ø17       IF J = 4 THEN PRINT "VYRILLIEX OUTWORLDERS"
7Ø18       IF J = 5 THEN PRINT "A LONE SHIP, APPARENTLY UNDER",,
             "ROBOT CONTROL"
7Ø19       IF J = 6 THEN PRINT "A PARRALEXIAN ESCORT VESSEL"
7Ø2Ø       PRINT
7Ø25       PRINT
7Ø27       PRINT
7Ø3Ø       LET Z = 1          + (FOLK/ (RND(15) + 1)
7Ø4Ø       PRINT "THERE WERE * ";Z;" * PEOPLE KILLED"
7Ø45       PRINT
7Ø5Ø       LET ZZ = 25Ø + RND(25Ø)
7Ø6Ø       PRINT " * * * DAMAGE WAS  £";ZZ
7Ø62       PRINT
7Ø65       LET ZZZ = RND(3ØØ)
7Ø66       LET ZZZZ = RND(3ØØ)
7Ø67       PRINT "AND FOOD STOCKS HAVE FALLEN",,"BY * ";ZZZZ
7Ø69       LET FOOD = FOOD - ZZZZ
7Ø7Ø       LET FOLK = FOLK - Z
7Ø75       LET OXY = OXY - ZZZ
7Ø8Ø       LET CASH = CASH - ZZ
7Ø85       PRINT
7Ø9Ø       PRINT  ,,"PRESS N/L"
7Ø92       INPUT U$
7Ø95       CLS
71ØØ       RETURN

8Ø1Ø       PRINT A$
8Ø2Ø       PRINT "YOU RAN OUT OF OXYGEN IN YEAR * ";YEAR
```

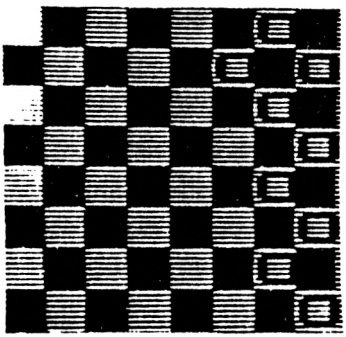
```
8040      GOTO 8020
8100      PRINT A$
8110      PRINT "FOOD SUPPLIES WERE EXHAUSTED IN YEAR * ";YEAR;
8120      GOTO 8100
8200      PRINT A$
8210      PRINT "THE TREASURY RAN DRY DURING YEAR * ";YEAR;" * ";
8220      GOTO 8210
8310      PRINT "YOUR POPULATION HAS FALLEN"
8315      LET FOLK = RND(26)
8320      PRINT "TO * ";FOLK;". DO YOU WANT TO"
8325      LET CASH = RND(300)
8330      PRINT "COMMIT SUICIDE PAINLESSLY "
8340      PRINT "NOW (1) OR AWAIT A SAD AND"
8350      PRINT "LINGERING DEATH? (2)"
8360      INPUT B
8365      CLS
8370      IF B =  1 THEN GOTO 8400
8385      PRINT "I HOPE YOU HAVE CHOSEN WELL"
8395      GOTO 20
8400      PRINT "GOOD BYE * ";
8410      GOTO 8400
```

CHESSBOARD NIM

You and the computer take it in turns to take pieces from the chessboard. The player who takes the last piece loses the game. The computer is programmed NOT to play a perfect game, to give you a chance to win now and again. If you want the ZX80 to win every time, and there is little fun in playing with it if you do, delete line 340.

```
10        LET S = 0
20        LET C = 0
30        LET Z = 20 + RND(11)
40        LET F = 2 + RND(3)
50        DIM A(32)
60        GOTO  380
70        IF S = 0 THEN PRINT "PIECES ON BOARD ";Z
80        IF S = 0 THEN PRINT "MAXIMUM NUMBER TO REMOVE ";F
90        IF C > 0 AND S = 0 THEN PRINT "YOU TOOK ";C;
                                  ", I TOOK ";D
100       FOR A = 0 TO 3
110       PRINT
120       FOR B = 0 TO 3
130       PRINT CHR$(A(29 + A - B*8));CHR$(128);
140       NEXT B
150       PRINT
160       FOR B = 0 TO 3
170       PRINT CHR$(128);CHR$(A(25 + A - B*8));
180       NEXT B
```





```
190      NEXT A
200      IF S = 1 THEN PRINT ,"YOU WIN"
210      IF S = 2 THEN PRINT ,"I WIN"
220      IF S > 0 THEN STOP
230      PRINT
240      PRINT "HOW MANY WILL YOU TAKE?"
250      INPUT C
260      IF C < 1 OR C > F THEN GOTO 250
270      LET Z = Z - C
280      IF Z > 0 THEN GOTO  310
290      LET S = 2
300      IF S  = 2 THEN GOTO 380
310      LET D = Z - 1 - ((Z - 1)/(F + 1))*(F + 1)
320      IF D = 0 AND NOT Z = 1 THEN LET D = RND(F)
330      IF NOT D < Z THEN GOTO 320
340      IF Z < F + 2 AND RND(4) = 4 THEN LET
               D = D + RND(2) - RND(2)
345 IF D> F THEN GOTO 310
350      IF D = 0 THEN LET D = 1
360      LET Z = Z - D
370      IF Z = 0 THEN LET S = 1
380      FOR A = 1 TO Z
390      LET A(A) = 52
400      NEXT A
410      FOR A = Z + 1 TO 32
420      LET A(A) = 0
430      NEXT A
440      CLS
450      GOTO 70
```

BOMB

You are searching on a solid black plane (a 9 x 9 grid of CHR$(128)'s) for a bomb, armed only with your keyboard and NEWLINE. A 'bomb detector' — a number which appears below the large black square — gives you clues to tell you where you are in relation to the bomb. You'll have to learn how to interpret the detector's output. You move by hitting the keys 5, 6, 7 or 8 before NEWLINE, and you move in the direction of the arrows on these keys. When you find it, an inverse "B" will appear where the bomb was hidden. This 1K program also gives you a readout of how long it took you to find it. If you are converting this program to run on a new ROM machine, you can change lines to accept an INKEY$ input (you will need a PAUSE 40) as well to save you having to press NEWLINE after each guess.

```
10      LET S = 0
20      LET A = 1 + RND(8)
30      LET B = RND(9)
40      LET K = 16396
50      LET Q = 128
60      FOR Z = 1 TO 10
70      PRINT ,,,,
80      NEXT Z
90      FOR X = 2 TO 10
100     FOR Y = 1 TO 10
110     POKE Y * 33 + X + PEEK(K) + PEEK( K + 1)*256, Q
120     NEXT Y
130     NEXT X
140     LET C = 1
150     LET D = 1
160     POKE C * 33 + D + 1 + PEEK(K) + PEEK(K + 1)*256, 189
170     LET S = S + 1
180     INPUT E$
190     POKE C * 33 + D + 1 + PEEK(K) + PEEK (K + 1)*256, Q
200     IF E$ = "7" THEN LET C = C - 1
210     IF E$ = "5" THEN LET D = D - 1
220     IF E$ = "6" THEN LET C = C + 1
230     IF E$ = "8" THEN LET D = D + 1
240     IF C<1 THEN LET C = 1
250     IF C>9 THEN LET C = 9
260     IF D<1 THEN LET D = 1
270     IF D>9 THEN LET D = 9
280     IF A = C AND B = D THEN GOTO 310
```

```
29Ø     POKE 187 + PEEK(K) + PEEK(K + 1)*256, ABS(B - D)+
                  ABS(A - C) + 156
3ØØ     GOTO 16Ø
31Ø     PRINT "SUCCESS AT * ";C,D;" * IN * ";S;" * TRIES"
32Ø     POKE C *33 + D + 1 + PEEK(K) + PEEK (K + 1)*256, 167
```

BLACKJACK

John Scarne, in his authoritative SCARNE'S ENCYCLOPEDIA OF GAMES, says Blackjack is "the most widely played banking card game in the world". It is relatively simple to play: The players try to get as close as possible to a total of 21, without exceeding 21. Aces count as either 1 or 11, and Kings, Queens, and Jacks each count as 10. This program automatically assigns a value of 1 to an Ace if counting it as 11 would force the total over 21. The human player always goes first in this ZX80 version of the game. After each card is handed out, you have the option of taking another one, or "standing", that is staying as you are (see line 8Ø). The round is a draw if both of you reach the same total, and it is less than 21. If you "bust", that is you exceed 21, the ZX80 wins that round automatically. The PRINT lines in this 2K program are a good example of ZX80 arrogance.

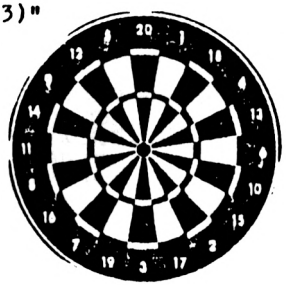
```
1Ø      GOTO 16Ø
2Ø      LET CARD = RND(11)
3Ø      IF CARD = 11 AND D + CARD> 21 THEN LET CARD = 1
35      LET D = D + CARD
4Ø      RETURN

5Ø      LET CARD = RND(11)
6Ø      IF CARD = 11 AND B + CARD> 21 THEN LET CARD = 1
65      LET B = B + CARD
7Ø      RETURN

8Ø      PRINT " * * * ANOTHER CARD (1) OR WILL",,"YOU STAND (Ø)?"
9Ø      INPUT G
1ØØ     RETURN

11Ø     PRINT ,,,,, " * * * ANOTHER GAME, CARD-SHARP?"
12Ø     INPUT A$
13Ø     CLS
14Ø     IF NOT A$ = "NO" THEN RUN
15Ø     STOP
```

```
16Ø     LET D = Ø
17Ø     LET B = Ø
18Ø     GOSUB 2Ø
19Ø     LET H = CARD
2ØØ     GOSUB 2Ø
21Ø     LET A = CARD
22Ø     GOSUB 5Ø
23Ø     LET E = CARD
24Ø     GOSUB 5Ø
25Ø     LET F = CARD

26Ø     LET B$ = "THE ZX8Ø HAS * ";
27Ø     LET C$ = "THE HUMAN HAS * ";
28Ø     PRINT ,B$;H
29Ø     PRINT ,C$;E;" * AND * ";F
3ØØ     PRINT ,,"TOTALLING * "; E + F
31Ø     LET D = H + A
32Ø     LET B = E + F
33Ø     IF B = 21 THEN GOTO 44Ø
34Ø     GOSUB 8Ø

35Ø     IF G = 1 THEN GOTO 49Ø
36Ø     CLS
365     IF D <17 THEN GOTO 53Ø
37Ø     IF NOT D = 21 THEN PRINT ,B$;D
38Ø     IF NOT B = 21 THEN PRINT ,C$;B
39Ø     IF B = D AND NOT B = 21 THEN PRINT ,"SO THIS ROUND
          IS A DRAW"
4ØØ     IF D = 21 AND NOT B = 21 THEN PRINT B$;"BLACKJACK..."
4Ø5     IF B > 21 THEN PRINT C$;"BUSTED",,"SO ZX8Ø WINS..."
41Ø     IF D < B AND NOT D > 21 THEN PRINT " * * *  ZX8Ø DESTROYS
        HUMAN WITH A ",,"BRILLIANT DISPLAY OF",,"CARD PLAYING"
415     IF D > 21 THEN PRINT B$;"BUSTED"
42Ø     IF D < 21 OR (B > D AND NOT B > 21) THEN PRINT " * * *
          YOU HAVE WON SOMEHOW...",,"LUCK, I GUESS"
43Ø     GOTO 11Ø
44Ø     PRINT C$;"BLACKJACK"
46Ø     IF NOT D = 21 THEN GOTO 37Ø
47Ø     PRINT "BUT SO HAS THE CLEVER COMPUTER,",,"SO ITS A DRAW"
48Ø     GOTO 11Ø
49Ø     GOSUB 5Ø
5ØØ     PRINT C$;CARD;" * TOTAL: * ";B
51Ø     IF B> 21 THEN GOTO 4ØØ
52Ø     GOTO 34Ø
53Ø     PRINT B$;D
535     INPUT U$
54Ø     GOSUB 2Ø
56Ø     PRINT B$;CARD
57Ø     PRINT "SO ITS TOTAL IS * ";D
575     INPUT U$
58Ø     IF D> 21 THEN GOTO 42Ø
59Ø     IF D < 17 THEN GOTO 54Ø
6ØØ     GOTO 37Ø
```

DARTS

This 1K program shows the REM statement being used to hold DATA for subsequent READing, thus substituting for the absent READ/DATA function in ZX80 BASIC. This game allows two players to choose from three shots to build up a total of 250. The result of your choice is not entirely random, and quite specific strategies can be developed which will give different results. Play it a number of times to determine your own strategy before challenging a friend. The score for player one is stored in address 16541 (on the old ROM ZX80), and player two's score is 16452. These are reset to zero, by lines 20 and 30, every time a new game is played. These are the only two addresses which are POKEd with new values, so these are the only two which have to be reset. The others -- the values of the "darts throws" -- are only PEEKed. One is added to the values obtained from the REM statement, and these are multiplied by 10 in the PRINT lines 1050, 1060 and 1070.

```
10      REM  8 shiftQ   shiftW  shiftE  shiftW  space   shiftW
             space    shiftW   shiftE  shiftE  space   shiftE
             space    shiftE   shiftD  shiftE  4spaces
20      POKE 16451, 0
30      POKE 16452, 0
40      PRINT "WHICH SHOT PLAYER 1 (1 TO 3)"
50      LET Z = 1
60      INPUT A
70      IF A< 1 OR A > 3 THEN GOTO 60
80      LET A = A + 1
90      CLS
100     GOSUB 1000
110     CLEAR
120     PRINT
130     PRINT "WHICH SHOT PLAYER 2? (1 TO 3)"
140     INPUT A
150     LET Z = 2
160     IF A < 1 OR A> 3 THEN GOTO 140
170     LET A = A + 1
180     PRINT
190     GOSUB 1000
200     PRINT
210     PRINT ,"N/L FOR NEXT THROW"
220     INPUT A$
230     IF NOT A$ = "" THEN STOP
240     CLS
250     GOTO 40
```



```
1000    LET C = 1 + RND(12)
1010    IF A = PEEK(16426 + C) THEN LET J = PEEK (16450 + Z)
1020    IF NOT A = PEEK(16426 + C) THEN GOTO 1000
1030    POKE 16450 + Z, J + PEEK(16427 + C)
1040    IF PEEK(16427 + C ) = 6 THEN PRINT ,"--BULLSEYE--"
1050    PRINT "YOU GOT * ";10*PEEK(16427 + C);" * FROM
                                A * ";A - 1
1060    IF Z = 1 THEN PRINT ,"TOTAL, PLAYER 1: * ";
                      10*PEEK(16451)
1070    IF Z = 2 THEN PRINT ,"TOTAL, PLAYER 2: * ";
                      10*PEEK(16452)
1080    IF 10*PEEK(16450 + Z) > 250 THEN GOTO 2000
1090    RETURN
2000    CLS
2010    PRINT "PLAYER * ";Z;" * IS THE WINNER"
2020    PRINT ,,"PLAYER 1: * ";10*PEEK(16451)
2030    PRINT ,,"PLAYER 2: * ";10*PEEK(16452)
2040    CLEAR
2050    PRINT "WHAMMO...";
2060    GOTO 2050
```

MORDECHAI-MIND

This game is marketed under the trade name MASTER-MIND by Invicta, who bought the rights to the game from an amateur mathematician, Mordechai Meirovich, in 1971. The game has been popular in England for centuries under the name BULLS AND COWS. The principle is simple. The computer picks a four-digit code, using the numbers 1 to 9, and never repeating the same digit within the code. You input your guess for the number, as a four-digit number, then press NEWLINE. A correct digit in the correct location will gain you a "black", a correct digit in the wrong position scores a "white". You have just 10 guesses to crack the code. Do not use the same digit twice in a single guess, or you'll confuse the poor little ZX80.

```
10      DIM C(4)
20      DIM G(4)
30      LET C(1) = RND(9)
40      FOR Z = 2 TO 4
50      LET C(Z) = RND(9)
60      FOR J = 1 TO Z - 1
70      IF C(J) = C(Z) THEN GOTO 40
80      NEXT J
90      NEXT Z
```

```
100      FOR G = 1 TO 10
110      INPUT A
120      LET A1 = A
130      FOR Z = 1 TO 4
140      LET G(Z) = A - 10*(A/10)
150      LET A = A/10
160      NEXT Z
170      LET B = 0
180      FOR Z = 1 TO 4
190      LET W = 0
200      IF NOT C(Z) = G(Z) THEN GOTO 230
210      LET B = B + 1
220      LET G(Z) = 0
230      NEXT Z
240      FOR Z = 1 TO 4
250      IF G(Z) = 0 THEN GOTO  300
260      FOR J = 1 TO 4
270      IF NOT C(Z) = G(J) THEN GOTO  290
280      LET W = W + 1
290      NEXT J
300      NEXT Z
310      PRINT A1;" * SCORED * * ";CHR$(B + 156);" * BLACK";
320      IF B = 1 THEN PRINT " ";
330      IF NOT B = 1 THEN PRINT "S";
340      PRINT " * * * ";CHR$(W + 156);" * WHITE";
350      IF NOT W = 1 THEN PRINT "S";
360      PRINT
370      IF B = 4 THEN PRINT "YOU GUESSED IT * ";
380      IF B = 4 THEN GOTO 410
390      NEXT G
400      PRINT "THE CORRECT CODE WAS * ";
410      FOR Z = 1 TO 4
420      PRINT C(5 - Z);
430      NEXT Z
```

CHALLENGE CHECKERS

The Victorians loved to play a variation of DRAUGHTS which they called REVERSI (a name also given to an early version of OTHELLO, but that's another story). This is the ZX80 and ZX81 version of the game. It's played on a standard draughts board. You start on the right hand side, the computer on the left. In CHALLENGE CHECKERS, you follow the ordinary rules of draughts, (diagonal moves only, capture by jumping over an opponent's piece into an empty square) with a few exceptions. There are no multiple jumps,

and no kings. As well, you can move backwards and forwards as you choose (as if you had a board of kings). But the major difference between CHALLENGE CHECKERS and draughts — and the whole point of the game — is the result of a capture. In ordinary draughts you take your opponent's piece off the board. In CHALLENGE CHECKERS the piece changes to become one of yours (which is why the Victorians called it REVERSI). The winner is the first player to "convert" eight pieces.

If you'd like to read more about turn-of-the-century variants of draughts, take a look at Roger Millington's splendid book GAMES AND PUZZLES FOR ADDICTS. The giant book PLAY THE GAME — OVER 40 GAMES FROM THE GOLDEN AGE OF BOARD GAMES compiled by Brian Lowe also shows how other games were developed from draughts.

TWO DIRE WARNINGS: Sometimes you'll find CHALLENGE CHECKERS locked into a I'LL TAKE ONE OF YOURS, THEN YOU TAKE IT BACK cycle. If this happens, move another piece, or you'll still be playing the game when your computer reverts to its constituent elements. The second warning: This game is highly addictive. Oh, and by the way, you need at least 4K to play this game.

```
10      GOTO 9000
20      GOSUB 5000
35      LET Q = 0
40      IF SI = 8 THEN LET Q = 1
50      IF SM = 8 THEN LET Q = 2
1000    IF Q > 0 THEN GOTO 5000
1010    PRINT
1020    IF A$ >  "" THEN PRINT "YOUR LAST MOVE WAS TO * ";A$
1030    PRINT "THIS MOVE: FROM? (LETTER, NUMBER)"
1040    INPUT B$
1050    PRINT B$;" *  TO?"
```

```
1060    INPUT C$
1070    LET A$ = C$
1080    FOR W = 1 TO 2
1090    IF W = 1 THEN LET E$ = B$
1095    IF W = 2 THEN LET E$ = C$
1100    LET Y(W) = -50*(E$ = "C4") -62*(E$ = "A2") -49*(E$ ="A4")
                   -36*(E$ = "A6") -23*(E$ = "A8")  -69*(E$ = "B1")
                   -56*(E$ = "B3") - 43*(E$ = "B5") - 63*(E$ =
                   "C2") - 37*(E$ = "C6") - 24*(E$ = "C8") - 70*
                   (E$ = "D1") - 57*(E$ = "D3") - 44*(E$ = "D5")
                   -31*(E$ = "D7") - 64*(E$ = "E2") - 51*(E$ =
                   "E4") - 38*(E$ = "E6") - 25*(E$ = "E8") -71*
                   (E$ = "F1") - 58*(E$ = "F3") - 45*(E$ = "F5")
                   -32*(E$ = "F7") - 65*(E$ = "G2") -52*(E$ =
                   "G4") - 39*(E$ = "G6") -26*(E$ = "G8") -72*
                   (E$ = "H1") - 59*(E$ = "H3") - 46*(E$ = "H5")
                   -33*(E$ = "H7") -30* (E$ = "B7")
1110    NEXT W
1120    LET A(Y(2)) = 1
1130    LET A(Y(1)) = 0
1140    IF ABS(Y(1) - Y(2)) >  7 THEN LET SM = SM + 1
1150    IF Y(1) - Y(2) = 12 THEN LET A(Y(1) - 6) = 1
1160    IF Y(1) - Y(2) = 14 THEN LET A(Y(1) - 7) = 1
1170    IF Y(2) - Y(1) = 12 THEN LET A(Y(2) - 6) = 1
1180    IF Y(2) - Y(1) = 14 THEN LET A(Y(2) - 7) = 1
1190    LET MOVE = 1
1200    GOSUB 5000

2000    REM COMPUTER JUMPS
2005    LET X = 0
2010    FOR Z = 23 TO 72
2020    IF NOT A(Z) = 9 THEN GOTO 2080
2040    IF A(Z + 14) = 0 AND A(Z + 7) = 1 THEN LET X = 14
2050    IF X = 0 AND A(Z + 12) = 0 AND A(Z + 6) = 1
                   THEN LET X = 12
2060    IF X = 0 AND A(Z - 14) = 0 AND A(Z - 7) = 1
                   THEN LET X =  -14
2070    IF X = 0 AND A(Z - 12) = 0 AND A(Z - 6) = 1
                   THEN LET X =  -12
2080    IF X = 0 THEN NEXT Z
2090    IF X = 0 THEN GOTO 3000
2100    LET SI = SI + 1
2105    LET A(Z) = 0
2110    LET A(Z + X) = 9
2120    LET A(Z + X/2) = 9
2130    GOTO 20

3000    REM COMPUTER MOVES SAFELY
3005    LET X = 0
3015    LET Y = 0
3030    LET Z = 22 + RND( 50 )
3040    LET Y = Y + 1
3050    IF Y < 50 AND NOT A(Z) = 9 THEN GOTO 3030
3060    IF X = 0  AND A(Z - 6) = 0 AND A(Z - 12) = 0
                   THEN LET X = - 6
3070    IF X = 0 AND A(Z - 7) = 0 AND A(Z - 14) = 0
                   THEN LET X = - 7
```

```
3080    IF X = 0 AND A(Z + 6) = 0 AND A(Z + 12) = 0
                THEN LET X = 6
3090    IF X = 0 AND A(Z + 7) = 0 AND A(Z + 14) = 0
                THEN LET X = 7
3100    IF X = 0 AND Y<50 THEN GOTO 3030
3110    IF X = 0 AND Y>49 THEN GOTO 4000
3120    LET A(Z) = 0
3130    LET A(Z + X) = 9
3140    GOTO 20

4000    REM COMPUTER MOVES RANDOMLY
4010    LET Y = 0
4020    LET Z = 22 + RND(50)
4030    LET Y = Y + 1
4040    IF Y  50 AND NOT A(Z) = 9 THEN GOTO 4020
4050    IF A(Z + 7) = 0 THEN LET X = 7
4060    IF X = 0 AND A(Z + 6) = 0 THEN LET X = 6
4070    IF X = 0 AND A(Z - 6) = 0  THEN LET X = -6
4080    IF X = 0 AND A(Z - 7) = 0 THEN LET X = -7
4090    IF X = 0 AND Y < 50 THEN GOTO 4020
4100    IF X = 0 AND Y > 49 THEN GOTO 4500
4110    LET A(Z + X) = 9
4120    LET A(Z) = 0
4130    GOTO 20
4500    LET Q = 2
4510    GOTO 40
```

(NOTE: EACH "/" below represents a shift A)

```
5000    REM PRINT BOARD
5002    LET U$ = ""
5005    CLS
5010    PRINT
5012    PRINT
5013    PRINT
5015    PRINT "ZX80 *  ;SI,,, "HUMAN * ";SM
5040    PRINT
5045    PRINT ," * * * * * 12345678"
5050    PRINT ," 4spaces shiftF  8shiftT  shiftD"
5060    PRINT ," * * * A";CHR$(130);"/";A(62);"/";A(49);"/";
                  A(36);"/";A(23);"shiftQ"

5070    PRINT ," * * * B";CHR$(130);A(69);"/";A(56);"/";A(43)
                  "/";A(30);"/ shiftQ"
5080    PRINT ," * * * C";CHR$(130);"/";A(63);"/";A(50);"/";
                  A(37);"/";A(24);"shiftQ"
5090    PRINT ," * * * D";CHR$(130);A(70);"/";A(57);"/";A(44)
                  "/";A(31);"/ shiftQ"
5100    PRINT ," * * * E";CHR$(130);"/";A(64);"/";A(51);"/";
                  A(38);"/";A(25);"shiftQ"
5110    PRINT ," * * * F";CHR$(130);A(71);"/";A(58);"/";A(45)
                  "/";A(32);"/ shiftQ"
```

```
5120    PRINT ," * * * G";CHR$(130);"/";A(65);"/";A(52);"/";
                  A(39);"/";A(26);" shiftQ"
5125    PRINT ," * * * H";CHR$(130);A(72);"/";A(59);"/";A(46);
                  "/";A(33);"/ shiftQ"
5130    PRINT ," 4spaces shiftR  8shiftG  shiftE"
5135    PRINT ," * * * * * 12345678"
5140    IF Q = 1 THEN PRINT ,,"I WIN"
5141    IF Q = 2 THEN PRINT ,,"YOU WIN"
5142    IF Q > 0 THEN STOP
5143    IF MOVE = 1 THEN PRINT ,,,,,"YOUR MOVE WAS TO * ";C$
5144    IF MOVE = 1 THEN INPUT U$
5145    IF MOVE = 1 AND U$ = "S" THEN STOP
5150    LET MOVE = 0
5160    RETURN
9000    DIM A(92)
9005    LET MOVE = 0
9010    LET A$ = ""
9015    LET Q = 0
9020    LET SI = 0
9030    LET SM = 0
9040    FOR A = 1 TO 92
9050    LET A(A) = 2
9060    NEXT A
9070    FOR A = 23 TO 39
9080    IF A = 29 OR A = 28 OR A = 27 OR A = 34 OR A = 35
            THEN GOTO 9100
9090    LET A(A) = 1
9100    NEXT A
9105    DIM Y(2)


9110    FOR A = 43 TO 52
9120    IF A = 47 OR A = 48 THEN GOTO 9140
9130    LET A(A) = 0
9140    NEXT A
9150    FOR A = 56 TO 72
9160    IF A = 60 OR A = 61 OR A = 66 OR A = 67 OR A = 68
                THEN GOTO 9180
9170    LET A(A) = 9
9180    NEXT A
9190    PRINT
9200    PRINT "WILL I HAVE THE FIRST"
9120    PRINT ,"MOVE? (Y OR N)"
9220    INPUT P$
9230    IF P$ = "Y" THEN GOTO 2000
9240    GOTO 20
```

ASCOT

Five tiny horses, convincingly disguised as the letters A to E, race up the 4K screen and then down again in ASCOT.

When they reach the finish line, the moving display freezes for a time, showing the horses, and giving each rider points for the way he has ridden. Once you've read the results (and collected your winnings from your friends), the display automatically "unfreezes" and a new race is underway. This program is an interesting example of full-screen graphics.

(Note: Lines 1Ø to 13Ø are the "Vasey moving display" which is used in certain other programs in this book. The first POKE address in line 3Ø should be 19ØØØ as given, for 4K programs. The GOTO destination in line 5Ø may be different in other programs in this book using this routine. The routine is copyright © P Vasey, 1981, and may not be used as, or as part of, any product or program offered for sale or publication.)

```
1Ø      REM CDEØØ6CDC2Ø5Ø12ØØ1D9CDC2Ø518Ø3CDADØ1Ø6Ø81ØFE2A1E4Ø23
            221E4Ø7CDEØØC823DBFE3E383223400Ø65E1ØFED3FE3EECØ6192A
            ØC4ØCBFCCDADØ13EF5Ø42BFD352318CA
2Ø      FOR A = Ø TO 67
3Ø      POKE 19ØØØ + A, 16*(PEEK(16427 + 2*A) -28) + PEEK
                             (16428 + 2*A) - 28
4Ø      NEXT A
5Ø      GOTO 9ØØØ
1ØØ     POKE 16414, T
11Ø     POKE 16415, 255
12Ø     LET XX = USR(19ØØØ)
13Ø     RETURN
3ØØ     LET A = A - 1 + RND(2)
31Ø     POKE 33*ABS(A - 18) + 1Ø + PEEK(G) + PEEK(H)*256,38
33Ø     LET B = B - 1 + RND(2)
34Ø     POKE 33*ABS(B - 18) + 13 + PEEK(G) + PEEK(H)*256,39
35Ø     LET C = C - 1 + RND(2)
36Ø     POKE 33*ABS(C - 18) + 16 + PEEK(G) + PEEK(H)*256,4Ø
37Ø     LET D = D - 1 + RND(2)
38Ø     POKE 33*ABS(D - 18) + 19 + PEEK(G) + PEEK(H)*256,41
39Ø     LET E = E - 1 + RND(2)
4ØØ     POKE 33*ABS(E - 18) + 22 + PEEK(G) + PEEK(H)*256,42
41Ø     GOSUB 1ØØ

2ØØØ    IF A> 35 OR B>35 OR C>35 OR D>35  OR E > 35 THEN
                    GOTO 95ØØ

3ØØØ    POKE 33*ABS(A - 18) + 1Ø + PEEK(G) + PEEK(H)*256,128
3Ø1Ø    POKE 33*ABS(B - 18) + 13 + PEEK(G) + PEEK(H)*256,128
3Ø2Ø    POKE 33*ABS(C - 18) + 16 + PEEK(G) + PEEK(H)*256,128
```

```
3Ø3Ø    POKE 33*ABS(D - 18) + 19 + PEEK(G) + PEEK(H)*256,128
3Ø4Ø    POKE 33*ABS(E - 18) + 22 + PEEK(G) + PEEK(H)*256,128
3Ø5Ø    GOTO 3ØØ

9ØØØ    LET XX = Ø
9ØØ5    CLS
9Ø1Ø    LET A = 1
9Ø15    LET B = 1
9Ø2Ø    LET C = 1
9Ø25    LET D = 1
9Ø3Ø    LET E = 1
9Ø35    LET G = 16396
9Ø4Ø    LET H = G + 1
9Ø45    LET A$ = CHR$(128);CHR$(128);CHR$(128);CHR$(128)
9Ø5Ø    LET T = 24Ø
9Ø6Ø    FOR F = 1 TO 18
9Ø7Ø    PRINT "shiftA  5spaces *";A$;A$;A$;A$;"* 5spaces shiftA"
9Ø8Ø    NEXT F
9Ø9Ø    GOTO 3ØØ
95ØØ    PRINT ,"THE RACE IS OVER"
951Ø    PRINT " * * * * FINAL POINTS: A * ";A;" * * * B * ";B
952Ø    PRINT ,"C * ";C;" * * * D * ";D; * * * E * ";E
9525    LET T = Ø
953Ø    GOSUB 1ØØ
955Ø    RUN 9ØØØ
```

ANTI-HANGMAN

This, as you've cleverly deduced from the title, is HANGMAN in reverse. You think of a word, and the computer tries to guess it. When you RUN this program, the computer will first ask you how many letters there are in the word. Then you'll see CHARACTER? on the screen. Input the character you wish to use for blank spaces in your word (+, *, - or whatever). The ZX80 will then think of a letter. If this letter is in your word, type the *number* of the letter in the word. That is, if your word is APPLE and the computer guesses E, respond by typing 5. If the letter is wrong, input Ø. After a correct letter, the computer will leave it there until you type Ø, to allow for double letters. So, if it thought P, you'd respond with 2, then NEWLINE, then 3, then NEWLINE, then Ø. This 1K game is great fun to play, and you'll find the long-suffering ZX80 has a much better chance of guessing your word within its 1Ø goes if you think of a long word.

```
10      REM ETAONRISHDLFCMUGYPWBJKQXVZ
20      LET L = 10
30      PRINT "LENGTH OF WORD?"
40      INPUT N
50      PRINT "CHARACTER?"
60      INPUT Q$
70      LET Q = CODE(Q$)
80      DIM A(26)
90      DIM C(N)
100     DIM G(N)
110     FOR Z = 1 TO 26
120     LET A(Z) = PEEK (16426 + Z)
130     IF Z < N + 1 THEN LET G(Z) = Q
140     NEXT Z
150     LET Z = RND(3)
160     LET A$ = CHR$ (A(Z))
170     FOR J = Z TO 25
180     LET A(J) = A(J + 1)
190     NEXT J
200     LET A = 0
210     CLS
220     PRINT ,
230     FOR Z = 1 TO N
240     PRINT CHR$(G(Z));
250     NEXT Z
260     PRINT
270     PRINT
280     PRINT ,"LIVES=";L
290     PRINT ,I GUESS * ";A$
300     INPUT B
310     IF B = 0 THEN GOTO 350
320     LET A = 1
330     LET G(B) = CODE(A$)
340     GOTO 210
350     LET F = 0
360     FOR Z = 1 TO N
370     IF G(Z) = Q THEN LET F = 1
380     NEXT Z
390     IF F = 0 THEN PRINT ,"I WIN"
400     IF F = 0 THEN STOP
410     IF A = 0 THEN LET L = L - 1
420     IF L> 0 THEN GOTO  150
430     PRINT ,"YOU WIN"
```

SNAIL RUN

Four tiny snails crawl (at an ever-increasing speed) across your screen from right to left. They carry numbers on their backs. Well, they don't really, but you'll see what we mean when you RUN the program. If you feel adventurous, you could, I suppose, even bet on the outcome of a race. The

routine can be used for tiny cars, or whatever, and if you have more than 1K, you can easily extend the program for eight or more snails or vehicles.

LINES 1Ø TO 4Ø Vasey moving display. First POKE address in line 3Ø is 17270.

```
5Ø      GOTO 2ØØ
1ØØ     POKE 16414, 2ØØ
11Ø     POKE 16415, 255
12Ø     LET XX = USR(17270)
13Ø     RETURN

2ØØ     LET C = Ø
21Ø     LET E = Ø
22Ø     LET F = Ø
23Ø     LET D = Ø
24Ø     LET C = C + RND(3) - 1
25Ø     LET D = D + RND(3) - 1
26Ø     LET E = E + RND(3) - 1
27Ø     LET F = F + RND(3) - 1
28Ø     FOR A = 1 TO 28 - C
29Ø     PRINT " * ";
3ØØ     NEXT A
31Ø     PRINT "shiftT 1 shiftA"
32Ø     PRINT
33Ø     FOR A = 1 TO 28 - D
34Ø     PRINT " * ";
35Ø     NEXT A
36Ø     PRINT "shiftT 2 shiftA"
37Ø     PRINT
38Ø     FOR A = 1 TO 28 - E
39Ø     PRINT " * ";
4ØØ     NEXT A
41Ø     PRINT "shiftT 3 shiftA"
42Ø     PRINT
43Ø     FOR A = 1 TO 28 - F
44Ø     PRINT " * ";
45Ø     NEXT A
46Ø     PRINT "shiftT 4 shiftA"
47Ø     IF E > 26 OR D > 26 OR C > 26 OR E > 26 THEN STOP
48Ø     GOSUB 1ØØ
49Ø     CLS
5ØØ     GOTO 24Ø
```

JOYBOX

JOYBOX is a 1K fruit machine that behaves just like the ones you see lurking in pubs, except the money you win and lose goes nowhere except into the ZX80's variables store.

This program pays out with about the same frequency as a real machine, with the contents of the reels, and the chance with which they come up, stored in the REM statement in line 10. Make sure you input the program exactly as listed, or you'll end up with some strange fruit. The score reel is the middle one of the three displayed, except for + BAR + pays out when there are three in a row in any direction. You have to pay for each go. Input 0, then NEWLINE to pull the handle. If the HOLD option comes up, input the number(s) of the reel(s) you want to HOLD, with NEWLINE in between each one, then 0 to get things underway again.

```
10      REM XCHERRYLEMON..PLUM.ORANGE*BELL*0000000111111
          2222223333344
20      DIM A(8)
30      DIM H(5)
40      LET A = 200
50      LET B = 0
60      FOR Z = B TO 5
70      LET H(Z) = 0
80      NEXT Z
90      FOR Z = 0 TO 8
100     IF H(Z - 3*(Z/3)) = 1 THEN GOTO 120
110     LET A(Z) = PEEK(16457 + RND(25))- 28
120     IF A(Z) = 4 THEN LET H( Z - 3*(Z/3) + 3) = 1
130     NEXT Z
140     LET W = 0
150     IF A(3) = A(4) THEN LET W = 10
160     IF A(4) = A(5) AND W = 10 THEN LET W = 10*A(3) + 10
170     IF H(3)*H(4)*H(5) = 1 THEN LET W = 100
180     LET A = A + W - 5
190     FOR Z = 0 TO 2
200     LET H(Z) = 0
210     NEXT Z
220     LET H = 0
230     IF A < 60 + RND(280) THEN LET H = 1
240     CLS
250     PRINT
260     FOR Z = 0 TO 8
270     PRINT " * * * ";
280     FOR J = 2 TO 7
290     PRINT CHR$(PEEK(16426 + 6*A(Z) + J);
300     NEXT J
310     IF 3*((Z + 1)/3) = Z + 1 THEN PRINT
320     IF 3*((Z + 1)/3) = Z + 1 THEN PRINT
330     NEXT Z
340     PRINT
350     FOR Z = 0 TO 2
360     IF H(Z) = 1 THEN PRINT " * * *    +HELD+";
370     IF H(Z) = 0 THEN PRINT " 9 spaces";
```

```
38Ø     NEXT Z
39Ø     FOR Z = 1 TO 5
4ØØ     PRINT
41Ø     NEXT Z
42Ø     IF W > Ø THEN PRINT ,"PAYS * ";W
43Ø     PRINT ,"YOU NOW HAVE * ";A;"P"
44Ø     IF H = 1 THEN PRINT "HOLD OR * ";
45Ø     PRINT "START"
46Ø     INPUT N
47Ø     LET B = 3
48Ø     IF H*N = Ø THEN GOTO 6Ø
49Ø     LET H(N - 1) = 1 - H(N -1)
5ØØ     GOTO 24Ø
```

CRAPS

In THE COMPLETE BOOK OF DICE GAMES, Skip Frey describes Craps as "the premier dice game". According to Mr Frey, "it is played everywhere from back alleys to posh casinos in Las Vegas and Monte Carlo". Despite this glowing description, it can become a very dull game indeed when played with a computer. After all, to play the game you just roll dice, and if you have your trust ZX80 doing this for you, there isn't much else to do. Therefore, we've jazzed up the program a bit, to save you falling asleep at the keyboard. This Craps program gives you a starting stake of £20, and then adds to it, or takes away, in accordance with your luck with the dice.

OFFICIAL TERMS FOR CRAPS:

NATURAL —	A 7 or an 11 on the first roll is a NATURAL. Roll this, and you win.
CRAPS —	A 2, 3 or 12 on the first roll is CRAPS. Roll this, and you've lost.
POINT —	A 4, 5, 6, 8, 9 or 10 on the first roll becomes your POINT. In the program, the variable E is your point.

If you don't roll a natural or craps, you continue to roll until you "make your point". In this game, see line 270, you win the grand sum of £25 if you make your point. However, and this is a big however, if you throw a 7 before you make your point, you lose (line 120). The program subtracts £3, plus a pound for every roll of the dice you've made in that game. So long as you manage to end a game with £1 or more, the ZX80 will offer you a new game on this 1K program.

```
5       LET M = 20
10      LET A = 0
20      LET E = 0
30      PRINT ,"PRESS N/L TO ROLL"
40      INPUT A$
50      GOSUB 320
60      LET B = RND(6)
70      LET C = RND(6)
80      LET D = B + C
90      LET A = A + 1
95      IF A = 6 THEN CLS
97      IF A = 6 THEN LET A = 2
100     IF A = 1 THEN GOTO 150
110     IF D = E THEN GOTO 270
120     IF D = 7 THEN GOTO 300
130     PRINT "THE DICE CAME UP *";B;"*";C,"TOTAL *";D
140     GOTO 40
150     IF D = 7 OR D = 11 THEN GOTO 190
160     IF D < 4 OR D = 12 THEN GOTO 210
170     LET E = D
180     GOTO 130
190     PRINT "YOU ROLLED *";D;"* SO YOU WIN"
195     LET M = M + 5 + A
200     GOTO 220
210     PRINT "FATE GAVE YOU *";D;"* SO YOU LOSE"
220     PRINT "YOUVE GOT £";M
222     IF M < 1 THEN STOP
225     PRINT ,,,,,"ANOTHER GAME?"
230     INPUT A$
240     CLS
250     IF NOT A$ = "NO" THEN GOTO 10
260     STOP
270     PRINT "THAT TIME YOU GOT *";D
280     LET M = M + 5 + A
290     GOTO 220
300     PRINT "FOOL, YOU BLEW IT BY ROLLING *";D
305     LET M = M - 3 - A
310     GOTO 220
320     FOR F = 1 TO 100
330     NEXT F
340     RETURN
```

NOGOMOKU

This version of the allegedly oriental game of GOMOKU is called *NO*GOMOKU because it does not play particularly well (something of an understatement!). However, it is an interesting game, in which the entire board and state of play is stored in a REM statement, which changes as play progresses. The aim of this 1K (on the old ROM, it needs more memory on the new ROM machines) NOGOMOKU is to get five in a row, either horizontally, vertically or diagonally. There is no facility in the program for the computer to know when the game is over, nor who has won.

You can, at any time, terminate play by entering an "X" when the computer gives you a move prompt. Just input the number, as indicated below, relating to the square on which you wish to place your piece. If you get tired of matching your wits with the ZX80, you can change the game to AUTONOGOMOKU by deleting line 2Ø, changing line 15Ø to LET A = RND(49) and line 14Ø to LET C = RND(1Ø). Then add: 175 INPUT A$. Examine the board after each two moves, and let the winner be the first to get three in a row in any direction.

The key for moves:

1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	32	33	34	35
36	37	38	39	40	41	42
43	44	45	46	47	48	49

```
10      REM  (49 full stops)
20      RANDOMISE
30      FOR E = 16427 TO 16475
40      PRINT E, 18
50      NEXT E
60      LET D = 0
70      PRINT
80      PRINT
90      PRINT ,
100     FOR E = 16427 TO 16475
110     LET D = D + 1
120     PRINT CHR$(PEEK(E));" * ";
130     IF 7*(D/7) = D THEN PRINT ,,,,,,,
140     NEXT E
150     INPUT A
160     IF A<1 OR A>49 OR NOT PEEK(16426 + A) = 18
                              THEN GOTO 150
170     POKE 16426, 180
180     CLS
```

```
190     LET Z = 0
200     LET C = - 2 + RND(3)
210     LET F = Z + 16426 + A + C
220     IF NOT PEEK(F) = 18 THEN LET Z = Z + 1
230     IF Z = 1 THEN LET A = A + 7 - C
240     IF Z = 2 THEN LET A = A - 8
250     IF Z = 3 THEN LET A = A + 2
260     IF Z = 4  THEN LET A = A + 8
270     IF Z > 4 THEN GOTO 190
280     IF Z > 0 THEN GOTO 220
290     IF A + C < 1 OR A + C > 49 THEN GOTO 190
300     POKE  F, 189
310     RUN 60
```

HELEN'S BOOGIE

This 1K game is a kind of snakes and ladders on a board of 24 squares. You throw a die which determines how many squares along the board you advance.

```
1Ø      DIM A(2)
2Ø      RANDOMISE
3Ø      LET A(1) = Ø
4Ø      LET A(2) = Ø
5Ø      FOR Z = 1 TO 2
6Ø      GOSUB 1ØØØ
7Ø      IF J < Ø THEN LET J = Ø
8Ø      PRINT "PLAYER ";Z;", YOU THREW A ";J
9Ø      PRINT "YOU ARE NOW ON ";A(Z)
1ØØ     FOR W = Ø TO A(Z)
11Ø     PRINT CHR$(128);CHR$(129);"shiftA";
12Ø     NEXT W
13Ø     PRINT
14Ø     PRINT
15Ø     NEXT Z
16Ø     IF A(1) > 23  OR A(2) > 23 THEN GOTO 2ØØ
17Ø     INPUT A$
18Ø     CLS
19Ø     GOTO 5Ø
2ØØ     CLS
21Ø     PRINT "PLAYER "; -1*(A(1)> A(2)) - 2*(A(2)> A(1));
                         " WINS, WITH"
22Ø     PRINT ,ABS(A(1) - A(2))*173;" POINTS"
23Ø     POKE 16421, 24
24Ø     STOP

1ØØØ    FOR G = 1 TO RND(25)
1Ø1Ø    LET J = RND(6) - RND(2)
1Ø2Ø    NEXT G
1Ø3Ø    LET A(Z) = A(Z) + J
1Ø4Ø    IF A(Z) = 5 OR A(Z) = 17 THEN GOTO 1ØØØ
1Ø5Ø    LET Y = -4*(A(Z) = 2 OR A(Z)= 14) + 2*(A(Z) = 3 OR A(Z)
                = 15) + 3*(A(Z) = 4 OR A(Z) = 16) +
                (A(Z) = 7 OR A(Z) = 19) + (RND(4))*
                (A(Z) = 1Ø OR A(Z) = 22)
1Ø6Ø    LET A(Z) = A(Z) + Y
1Ø7Ø    IF A(Z) < Ø THEN LET A(Z) = Ø
1Ø8Ø    RETURN
```

Siege

This is a very frustrating game which makes very effective use of the 1K on an old ROM ZX80. You are a soldier (actually, you're the letter X, but this is a ZX80, not an IBM

HORROR 2017) trying to wend your way through army territory (translation: Trying to move from the right to the left of the screen) without either landing on top of an enemy (a black blob) or accidentally bumping into one. But it is a game you cannot win. No sooner do you make it across the enemy territory when you're whisked back to the start, to begin your tireless trek again, through an ever-increasing enemy hoard. When you finally die, you'll be given a score which is related to how long you've survived. Anything higher than 132 is very good. You move by entering a 7, then NEWLINE to move up, 6 to move down and 5 to move forward.

```
10      GOSUB 190
20      IF PEEK (C*33 + D + 1 + PEEK(Z) + PEEK (Z + 1)*256) = 61
              OR PEEK (Y*33 + X + 1 + PEEK(Z) + PEEK (Z + 1)
              *256) = 128 THEN GOTO 170
30      POKE Y*33 + X + 1 + PEEK(Z) + PEEK(Z + 1)*256, 61
40      POKE C*33 + D + 1 + PEEK(Z) + PEEK(Z + 1)*256, 128
50      INPUT A$
60      LET K = K + 1
70      POKE Y*33 + X + 1 + PEEK(Z) + PEEK(Z + 1)*256, 0
80      IF A$ = "5" THEN LET X = X - 1
90      IF A$ = "7" THEN LET Y = Y - 1
100     IF A$ = "6" THEN LET Y = Y + 1
110     LET C = RND(8)
120     LET D = RND(30)
130     IF X< 2 THEN LET X = 31
140     IF Y< 2 THEN LET Y = 2
150     IF Y> 9 THEN LET Y = 8
160     GOTO 20
170     PRINT "YOUR SCORE: * ";K
180     STOP
190     FOR J = 1 TO 10
200     PRINT ,,,,
210     NEXT J
220     LET X = 31
230     LET Y = 2
240     LET Z = 16396
250     LET C = 1
260     LET D = 1
270     RETURN
```

BATTLE

This game pits your skill against five marauding black blobs. At the beginning of the game, you — and the lumbering black blobs — are placed randomly within a frame on the screen. You move by entering a letter (N for north, E to move east and so on or Q if you want to stop the game). If you wish to keep moving in the same direction, just press NEWLINE when the prompt appears. One of the black blobs will move each. You are the dollar sign ($) and your million, should you decide to accept it, is to try and wipe out all the black blobs by landing on top of them before you've used up your 20 moves.

However, when you get near a blob, you are in danger of it landing on top of you. You are safe so long as you stay more than three spaces away from a blob. If a blob lands on top of you, the game is generally over.

Your score is shown to the left of the frame (the top number), while the bottom number counts off your moves (showing the last digit only). The game is complicated by the fact that the blobs, once erased, refuse to stay erased, and will return at a random time to plague you again. If a blob lands on you, you are — as we've said — generally done for, but a benevolent line in the program (910)allows you to escape from time to time.

A further complication is caused by a coding trick which makes youvanish at random intervals, so you have to make your next move "blind". You will, however, always appear at the next move, unless you've been squashed.

```
10 - 130   VASEY MOVING DISPLAY  (A = 190000, line 50 GOTO 220)
200     POKE Y * 33 + X + 1 + PEEK(16396) + PEEK (16397)*256, K
210     RETURN
220     LET XX = 0
230     GOTO 9000
490     PRINT "DIRECTION? (N S E W)"
```

```
5ØØ     LET Y = 11
5Ø2     LET MOVE = MOVE + 1
5Ø3     IF MOVE = 2Ø THEN LET T = Ø
5Ø4     IF MOVE = 2Ø THEN GOSUB 1ØØ
5Ø5     IF MOVE = 2Ø THEN GOTO 95ØØ
51Ø     LET X = 3
52Ø     LET K = CODE(STR$(SI))+ 128
55Ø     GOSUB 2ØØ
56Ø     LET Y = 13
57Ø     IF MOVE < 1Ø THEN LET K = CODE(STR$(MOVE)) + 128
575     IF MOVE > 9  THEN LET K = CODE(STR$(MOVE - 9)) + 128
58Ø     GOSUB 2ØØ
6ØØ     INPUT B$
6Ø3     IF B$ = "" THEN LET B$ = C$
6Ø4     LET Y = A
6Ø5     LET K = Ø
6Ø6     LET X = B
6Ø7     GOSUB 2ØØ
6Ø8     IF B$ = "Q" THEN STOP
61Ø     IF B$ = "N" THEN LET A = A - 1
62Ø     IF B$ = "S" THEN LET A = A + 1
63Ø     IF B$ = "E" THEN LET B = B + 1
64Ø     IF B$ = "W" THEN LET B = B - 1
65Ø     IF B < 9 THEN LET B = 9
66Ø     IF B > 21 THEN LET B = 21
67Ø     IF A < 8 THEN LET A = 8
675     IF A > 2Ø THEN LET A = 2Ø
68Ø     FOR C = 1 TO 5
681     IF Y(C) = A AND X(C) = B THEN LET SI = SI + 1
685     NEXT C
69Ø     LET Y = A
695     LET C$ = A$
7ØØ     LET X = B
7Ø5     LET K = 13
71Ø     GOSUB 2ØØ
715     GOSUB 1ØØ
72Ø     LET E = RND(5)
73Ø     LET K = Ø
74Ø     LET Y = Y(E)
75Ø     LET X = X(E)
76Ø     GOSUB 2ØØ
79Ø     LET Y(E) = Y(E) + RND(3) - RND(3)
8ØØ     LET X(E) = X(E) + RND(3) - RND(3)
81Ø     IF  Y(E) < 9 THEN LET Y(E) = 9
82Ø     IF Y(E) > 16 THEN LET Y(E) = 16
83Ø     IF X(E) <  9 THEN LET X(E) = 9
84Ø     IF X(E) > 2Ø THEN LET X(E) = 2Ø
85Ø     LET Y = Y(E)
86Ø     LET X = X(E)
87Ø     LET K = 128
88Ø     GOSUB 2ØØ
89Ø     GOSUB 1ØØ
9ØØ     FOR C = 1 TO 5
91Ø     IF Y(C) = A AND X(C) = B AND RND(3) = 1 THEN GOTO 95ØØ
92Ø     NEXT C
93Ø     GOTO 5ØØ
```

```
9000    DIM Y(5)
9005    LET SI = 0
9010    LET MOVE = - 1
9015    DIM X(5)
9020    LET C$ = "E"
9025    LET T = 240
9030    FOR A = 1 TO 5
9035    PRINT ,,,,
9040    NEXT A
9050    PRINT ,"(15 shift W)",,
9060    FOR A = 1 TO 12
9070    PRINT , "(shift Q 13 spaces)";CHR$(130),,
9080    GOSUB 100
9090    NEXT A
9100    PRINT ,"(15 shift G)",,
9110    PRINT
9115    LET K = 128
9120    FOR A = 1 TO 5
9130    LET Y(A) = 7 + RND(10)
9140    LET X(A) = 8 + RND(12)
9142    LET Y = Y(A)
9145    LET X = X(A)
9150    GOSUB 200
9160    GOSUB 100
9170    NEXT A
9180    LET A = 7 + RND(10)
9185    LET Y = A
9190    LET B = 8 + RND(12)
9200    LET X = B
9210    GOSUB 200
9220    GOTO 490

9500    CLS
9505    PRINT  "THE BATTLE IS OVER"
9510    PRINT
9512    PRINT "SCORE * ";SI
9515    PRINT ,"IN * ";MOVE;" * MOVES"
9517    PRINT
```

LOGICA

LOGICA lets you zero in on a target number chosen between one and 60, by giving you clues in the form of letters of the alphabet. Play this one a few times, and you'll learn how to interpret the computer responses to your guesses. Note how RUN n is used in this 1K game, and how the "secret number" and other information is stored in a non-volatile manner in the REM statement.

```
10      REM ???
20      POKE 16427, 1
30      RANDOMISE
35      POKE 16428, RND(60)
40      GOTO 120
50      PRINT , PEEK (16429);" * IS WRONG"
60      PRINT
70      PRINT " * * HERE IS A HINT:"
75      PRINT
80      FOR A = 1 TO ABS(PEEK(16428) - PEEK(16429))/2
90      PRINT CHR$(A + 37),
100     NEXT A
110     PRINT
120     PRINT " * * OK BEACH BOY, WHAT NUMBER,"
130     PRINT " * * BETWEEN 1 AND 60, AM I"
140     PRINT " * * HOLDING IN MY Z80 CHIP?"
150     PRINT
170     PRINT " * * THIS IS GUESS NUMBER * ";PEEK (16427)
180     POKE 16427, PEEK(16427) + 1
190     INPUT B
195     CLS
200     IF B = PEEK(16428) THEN RUN 300
210     POKE 16429, B
220     GOTO 50
300     CLS
310     PRINT ,"YOU ARE RIGHT"
320     PRINT
330     PRINT "I WAS THINKING OF * ";PEEK(16428)
340     PRINT
350     PRINT "YOU GOT IT IN JUST * ";PEEK(16427);" * GUESSES"
360     INPUT A$
370     CLS
380     IF A$ = "" THEN RUN
```

FALLEN COMRADES

FALLEN COMRADES is essentially a 1K Russian Roulette program for three people. Enter three names, and then leave your life, and the lives of your friends, in the hands of the Sinclair random number generator. If you've got more memory, you can easily expand the game for a whole baseball team of comrades.

```
10      RANDOMISE
20      LET E$ = " * SURVIVES"
30      PRINT "NAME OF PLAYER 1?"
40      INPUT A$
50      PRINT ,"AND 2?"     (note: get the AND from the 'shift
60      INPUT B$
70      PRINT ,"AND 3?"
80      INPUT C$
```

```
9Ø      LET Z = Ø
1ØØ     CLS
11Ø     LET Z = Z + 1
12Ø     PRINT
13Ø     PRINT ,"PRESS NEWLINE TO FIRE"
14Ø     INPUT D$
15Ø     PRINT
16Ø     PRINT
17Ø     PRINT A$, B$, C$
18Ø     PRINT
19Ø     IF A$ = "" THEN GOTO 24Ø
2ØØ     LET A = RND(1Ø)
21Ø     IF NOT A = 5 THEN PRINT "CLICK";
22Ø     IF A = 5 THEN PRINT "BANG";
23Ø     IF A = 5 THEN LET A$ = ""
24Ø     IF B$ = "" THEN PRINT ,
25Ø     IF B$ = "" THEN GOTO 3ØØ
26Ø     LET B = RND(1Ø)
27Ø     IF NOT B = 5 THEN PRINT,"CLICK";
28Ø     IF B = 5 THEN PRINT ,"BANG";
29Ø     IF B = 5 THEN LET B$ = ""
3ØØ     IF C$ = "" THEN GOTO 35Ø
31Ø     LET C = RND(1Ø)
32Ø     IF NOT C = 5 THEN PRINT ,"CLICK"
33Ø     IF C = 5 THEN PRINT ,"BANG"
34Ø     IF C = 5 THEN LET C$ = ""
35Ø     PRINT
36Ø     PRINT
37Ø     IF A$ = "" AND B$ = "" THEN PRINT C$;E$
38Ø     IF A$ = "" AND C$ = "" THEN PRINT B$;E$
39Ø     IF B$ = "" AND C$ = "" THEN PRINT A$;E$
4ØØ     IF (CODE(A$) + CODE(B$) = 2) OR (CODE (A$) + CODE(C$) = 2)
            OF (CODE (B$) + CODE(C$) = 2) THEN STOP
41Ø     PRINT
42Ø     PRINT ,"THIS IS ROUND NUMBER * ";Z
43Ø     INPUT Z$
44Ø     GOTO 1ØØ
```

VENUS LANDER

Forget about lunar landers. This one is, for some reason, set above misty Venus. Once you've managed to land your tumbling space craft successfully a number of times, make things more difficult by reducing your starting fuel (line 33Ø) or by changing your initial speed (line 34Ø). Venus Lander runs in 1K on an old ROM ZX80.

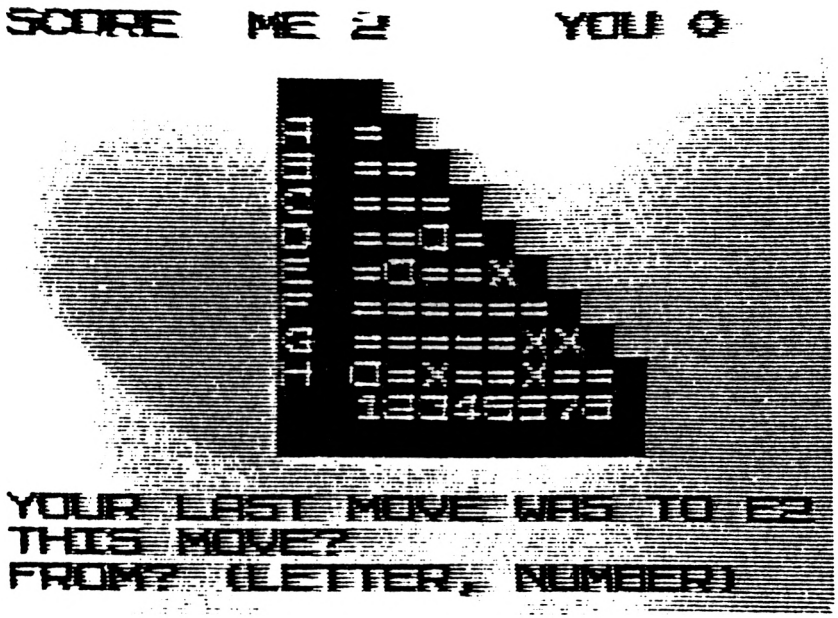
```
1Ø      GOSUB 32Ø
2Ø      CLS
3Ø      PRINT ,"HEIGHT","FUEL","SPEED"
35      PRINT ,H,F,S
4Ø      FOR A = 1 TO 16 - H/1ØØØ
```

```
50      PRINT
60      NEXT A
70      FOR B = 1 TO 11 + RND(5)
80      PRINT " * ";
90      NEXT B
100     LET J = RND(2)
110     IF J = 1 THEN PRINT "(shift F shift G shift D)"
120     IF J = 2 THEN PRINT "(shift R shift T shift E)"
130     FOR A = 16 - H/100 TO 16
140     PRINT
150     NEXT A
160     PRINT "(random set of 32 graphics symbols to represent
                    the surface of Venus)"
170     INPUT T
180     IF F - T < 1 THEN LET T = 0
190     LET S = S + S/10 + 5 - T
200     LET H = H - S
210     IF H > 1600 THEN GOTO 290
220     LET F = F - T/3
230     IF H> 0 THEN GOTO 20
240     IF S > 10 THEN GOTO 270
250     PRINT "SUCCESSFUL LANDING * ";F*23;" * POINTS"
260     STOP
270     PRINT "CRASH LANDING, FORMING * "; S * RND(7),
                    "METRE CRATER"
280     STOP
290     CLS
300     PRINT ABS(S);" - ESCAPE VELOCITY space shift F shift G
            shift D space";
310     GOTO 300
320     LET H = 1500 + RND(100)
330     LET F = 90 + RND(75)
340     LET S = 10 + RND(10)
350     RETURN
```

BERMUDA TRIANGLE

BERMUDA TRIANGLE is an excellent 4K program to demonstrate "machine intelligence". BERMUDA TRIANGLE owes debts to both chess and draughts/checkers...and also, of course, to that famous holiday resort where people, planes and ships have mysteriously vanished. You can move one square at a time in any direction on the triangular board. You move by just inputting the piece you want to move, in the form B4 then NEWLINE, then the square, say B3, you want to move to. When the screen clears, you'll see your piece in its new

position (you are the letter "O") and the computer's response (it is the "X"). You capture by landing on top of the computer's pieces. First player to capture four of the opponent's pieces wins. Although it takes a while to set up



the board at the start of the game, the computer's response time is generally very quick. Few moves should take more than 20 seconds, and most will take less.

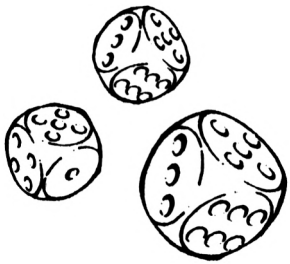
```
10      GOSUB 9000
500     CLS
1000    PRINT
1010    IF SI = 4 THEN LET Q = 1
1013    IF SM = 4 THEN LET Q = 2
1015    PRINT "SCORE * * ME * ";SI, "YOU * ";SM
1020    PRINT
1030    PRINT ,CHR$(128);CHR$(128);CHR$(128)
1040    FOR X = 0 TO 7
1050    PRINT ,CHR$(X + 166);CHR$(128);
1060    FOR J = 0 TO X
1070    PRINT CHR$(A(80 - 9*X - J));
```

```
1080    NEXT J
1090    PRINT CHR$(128)
1100    NEXT X
1102    PRINT ,CHR$(128);
1105    FOR J = 0 TO 8
1106    IF J = 0 THEN PRINT CHR$(128);
1107    IF J > 0 THEN PRINT CHR$( J + 156);
1108    NEXT J
1109    PRINT CHR$(128)
1115    PRINT ,
1120    FOR J = 1 TO 11
1125    PRINT CHR$(128);
1127    NEXT J
1128    PRINT
1130    IF Q = 17 THEN PRINT ,"I CONCEDE THE GAME"
1135    IF Q = 1 THEN PRINT ,,,"I WIN"
1137    IF Q = 2 THEN PRINT ,,,"YOU WIN"
1140    IF Q > 0 THEN STOP
1150    PRINT

2000    IF C$ >  "" THEN PRINT "YOUR LAST MOVE WAS TO _* ";C$
2010    PRINT "THIS MOVE?"
2012    FROM? (LETTER, NUMBER)"
2015    INPUT A$
2020    IF A$ = "S" THEN STOP
2040    PRINT A$;" _* TO? (LETTER, NUMBER)"
2050    INPUT B$
2060    LET C$ = B$
2070    LET G(1) = 451 - 9*CODE(A$) - CODE(TL$(A$))
2080    LET G(2) = 451 - 9*CODE(B$) - CODE(TL$(B$))
2090    IF A(G(2)) = 189 THEN LET SM = SM + 1
2100    LET A(G(1)) = 150
2110    LET A(G(2)) = 180

3000    REM COMPUTER JUMPS
3010    LET X = 0
3020    FOR Z = 10 TO 80
3025    IF NOT A(Z) = 189 THEN GOTO 3130
3030    IF A(Z + 9) = 180 THEN LET X = 9
3040    IF X = 0 AND A(Z + 8) = 180 THEN LET X = 8
3050    IF X = 0 AND A(Z - 8) = 180 THEN LET X = -8
3060    IF X = 0 AND A(Z + 10) = 180 THEN LET X = 10
3070    IF X = 0 AND A(Z - 10) = 180 THEN LET X = -10
3080    IF X = 0 AND A(Z - 9) = 180 THEN LET X = -9
3090    IF X = 0 AND A(Z + 1) = 180 THEN LET X = 1
3100    IF X = 0 AND A(Z - 1) = 180 THEN LET X = -1
3120    IF NOT X = 0 THEN GOTO 3150
3130    NEXT Z

3140    IF X = 0 THEN GOTO 3180
3150    LET A(Z) = 150
3160    LET A(Z + X) = 189
3170    LET SI = SI + 1
3175    GOTO 500
3180    LET Y = 0
3190    LET X = 0
3200    LET Z = 10 + RND(70)
```

```
3205    LET Y = Y + 1
3207    IF NOT A(Z) = 189 AND Y   40 THEN GOTO 3200     1
3210    IF A(Z) = 189 AND A(Z + 8) = 150 AND NOT A(Z + 16)
          = 180 AND NOT A(Z + 17) = 180 AND NOT A(Z + 7)
          = 180 AND NOT A(Z - 2) = 180 AND NOT A(Z - 1)
          = 180 AND NOT A(Z + 9) = 180 AND NOT A(Z + 18)
          = 180 THEN LET X = 8
3230    IF X = 0 AND A(Z) = 189 AND A(Z + 9) = 150 AND NOT
          A(Z + 10) = 180 AND NOT A(Z + 8) = 180  AND NOT
          A(Z + 18) = 180 AND NOT A(Z + 19) = 180 AND NOT
          A(Z + 17) = 180 AND NOT A(Z - 1)  = 180 AND NOT
          A(Z + 1)  = 180 THEN LET X = 9
3234    IF SI * SM = 0 AND RND(4) > 1 OR RND(3) > 1 THEN
          GOTO 3250
3235    IF Z < 20 THEN GOTO 3250
3237    IF X = 0 AND A(Z) = 189 AND A(Z - 8) = 150 AND NOT
          A(Z - 16) = 180 AND NOT A(Z - 17) = 180 AND NOT
          A(Z - 7)  = 180 AND NOT A(Z + 2)  = 180 AND NOT
          A(Z + 1)  = 180 AND NOT A(Z - 9)  = 180 AND NOT
          A(Z - 18) = 180 THEN LET X = - 8
3240    IF X = 0 AND A(Z) = 189 AND  A(Z - 9) = 150 AND NOT
          A(Z - 10) = 180 AND NOT A(Z - 8)  = 180  AND NOT
          A(Z - 18) = 180 AND NOT A(Z - 19) = 180 AND NOT
          A(Z - 17) = 180 AND NOT A(Z + 1)  = 180 AND NOT
          A(Z - 1)  = 180 THEN LET X = -9
3250    IF X = 0 AND Y <100 THEN GOTO 3200
3270    LET Y = 0
3290    LET Z = 10 + RND(70)
3295    LET Y = Y + 1
3300    IF NOT A(Z) = 189 AND Y < 100 THEN GOTO 3290
3320    IF A(Z + 8) = 150 THEN LET X = 8
3330    IF X = 0 AND A(Z - 8) = 150 THEN LET X = - 8
3340    IF X = 0 AND A(Z - 9) = 150 THEN LET X = - 9
3350    IF X = 0 AND A(Z + 10) = 150 THEN LET X = 10
3360    IF X = 0 AND A(Z - 10) = 150 THEN LET X = - 10
3370    IF X = 0 AND A(Z - 1)  = 150 THEN LET X = - 1
3380    IF X = 0 AND A(Z + 1)  = 150 THEN LET X = 1
3390    IF NOT X = 0 THEN GOTO 8000
3400    IF Y<100  THEN GOTO 3290

5000    REM ADMITS DEFEAT
5010    LET Q = 17

8000    LET A(Z) = 150
8010    LET A(Z + X) = 189
8020    GOTO 500
```

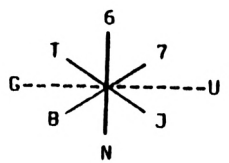
```
9000    DIM A(105)
9010    LET Q = 0
9020    LET SI = 0
9030    LET SM = 0
9040    DIM G(2)
9050    FOR B = 1 TO 105
9060    LET A(B) = 9
```

```
9070     NEXT B
9080     FOR B = 0 TO 7
9090     FOR R = 0 TO B
9100     LET A(80 - 9*B - R) = 150
9110     NEXT R
9120     NEXT B
9130     LET A(71) = 180
9140     LET A(35) = 180
9150     LET A(62) = 180
9160     LET A(53) = 180
9170     LET A(44) = 180
9180     FOR Z = 11 TO 15
9190     LET A(Z) = 189
9200     NEXT Z
9210     LET C$ = ""
9220     RETURN
```



PICASSO

Here's your chance to draw pretty pictures in 1K on your TV screen, using POKed graphics under cursor control. Pick the direction of the line you want to draw, then press NEWLINE. Change direction by entering the designated key (see below) then press NEWLINE over and over again. The ZX80 will automatically choose the most appropriate graphics character for the direction of the line you're drawing. The line will continue in the designated direction until you press another key. This table shows the directions away from the centre that you'll move if you input the key before pressing NEWLINE:




```
10       GOSUB 170
20       POKE Y*33 + X + 1 + PEEK(Z) + PEEK(Z + 1)*256, K
30       LET P$ = A$
40       INPUT A$
50       IF A$ = "" THEN LET A$ = P$
60       LET A = CODE(A$)
70       IF A = 34 OR A = 57 OR A = 35 THEN LET Y = Y - 1
80       IF A = 35 OR A = 51 OR A = 39 THEN LET Y = Y + 1
90       IF A = 47 OR A = 51 OR A = 47 THEN LET X = X + 1
```

```
100     IF A = 39 OR A = 44 OR A = 57 THEN LET X = X - 1
110     LET K = -8*(A = 39 OR A = 35) -136*(A = 47 OR A = 57)
                -7*(A = 34 OR A = 51 OR A = 44 OR A = 58)
120     IF X<2 THEN LET X = 2
130     IF X>31 THEN LET X = 31
140     IF Y<2 THEN LET Y = 2
150     IF Y>14 THEN LET Y = 14
160     GOTO 20
170     FOR J = 1 TO 15
180     PRINT ,,,,
190     NEXT J
200     LET X = 16
210     LET K = 6
220     LET Y = 7
230     LET Z = 16396
240     LET A$ = ""
250     RETURN
```

CHEMIN DE FER

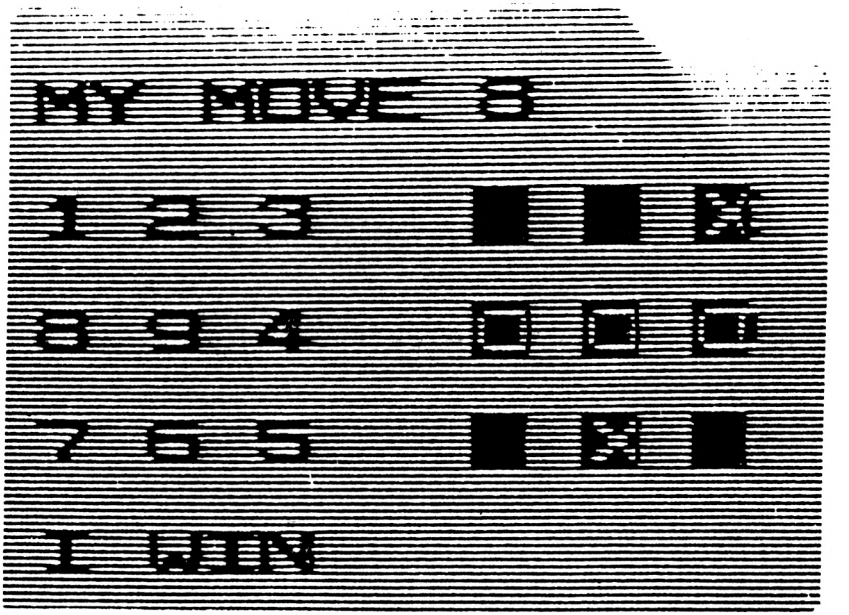
Baccarat was first introduced into France from Italy in about 1490, during the reign of Charles VIII. It is most unlikely, historians say, that Charles played it on a ZX80. The Italian game was called Baccara, and this game — Chemin De Fer — is a distant cousin of that old favourite. ZX80 Chemin De Fer is based on a dice version of the casino game which is usually played with cards. You and the ZX80 (the "banker") roll five dice each. If any die comes up 2 or 5, it *must* be rolled again. You add the pips on dice which did not come up 2 or 5, and then add to this the total of the pips from the dice you've rolled again. The aim is to get as close as possible to 9, or to get a two-digit number ending in 9. The program automatically strips a two-digit number down to its final digit. RUN it a few times, and you'll begin to see why this game is so popular. The program allows 9 winning games, with dead-heats (or "stand-offs") not counted. The winner is the player with the most games out of 9. The game as listed needs 2K. To run it in 1K, just add a CLS after each INPUT A$, except the last one, which already has one.

```
10      LET B1 = 0
20      LET P1 = 0
30      GOTO 340
40      LET D = 0
```

```
5Ø      LET C = Ø
6Ø      FOR G = 1 TO 5
7Ø      LET A = RND(6)
8Ø      IF A = 2 OR A = 5 THEN LET C = C + 1
9Ø      IF A = 2 OR A = 5 THEN LET A = Ø
1ØØ     PRINT A;" * ";
11Ø     LET D = D + A
12Ø     NEXT G
13Ø     PRINT
14Ø     PRINT D,
15Ø     IF D > 9 THEN LET D = D - 1Ø
16Ø     PRINT D
17Ø     IF D > 9 THEN LET D = D - 1Ø
18Ø     PRINT "TOTAL ON FIRST ROLL IS * ";D
19Ø     IF D = 9 THEN PRINT ,"LA GRANDE"
2ØØ     IF D = 8 THEN PRINT ,"LA PETITE"
21Ø     IF D = 7 THEN PRINT ,"NATURAL"
22Ø     IF C = Ø OR D = 7 OR D = 8 OR D = 9 THEN  RETURN
23Ø     PRINT "MUST ROLL * ";C;" * AGAIN"
24Ø     FOR A = 1 TO C
25Ø     LET E = RND(6)
26Ø     IF E = 2 OR E = 5 THEN LET E = Ø
27Ø     LET D = D + E
28Ø     NEXT A
29Ø     PRINT D,
3ØØ     IF D > 9 THEN LET D = D - 1Ø
31Ø     PRINT D
32Ø     IF D > 9 THEN LET D = D - 1Ø
33Ø     RETURN
34Ø     PRINT ,"BANKER"
35Ø     GOSUB 4Ø
36Ø     PRINT "FINAL TOTAL * ";D
37Ø     INPUT A$
38Ø     LET J = D
39Ø     PRINT ,"PLAYER"
4ØØ     GOSUB 4Ø
41Ø     INPUT A$
42Ø     PRINT ,"BANKER","PLAYER"
43Ø     PRINT,J,D
44Ø     IF J = D THEN PRINT ,"STAND-OFF
45Ø     IF J = D THEN GOTO 51Ø
46Ø     IF J > D THEN PRINT "BANKER";
47Ø     IF J > D THEN LET B1 = B1 + 1
48Ø     IF J < D THEN PRINT "PLAYER";
49Ø     IF J < D THEN LET P1 = P1 + 1
5ØØ     PRINT " * WINS"
51Ø     PRINT ,"TOTALS"
52Ø     PRINT ,B1,P1
53Ø     IF B1 + P1 = 9 THEN STOP
54Ø     INPUT A$
55Ø     CLS
56Ø     GOTO 34Ø
```

NOUGHTS & CROSSES

Nearly everyone knows how to play this game. Moving alternately, players put either a zero or a cross in a position on a 3 X 3 grid, trying to get three in a row in any direction. In



this game, the computer always goes first, and unsporting thing that it is, always hogs the middle square. There is no provision in this program for the player to win. The best you can do is draw. At first sight, NOUGHTS AND CROSSES might seem a relatively easy game to program, but in fact it is not simple at all. And it is more difficult to write a fallible program than it is to write a game which is virtually unbeatable. According to COMPUTERS, THEIR IMPACT AND USE, by Robert E Lynch and John R Rice, there are 362,800 possible different games of noughts and crosses. This program will play 40,320 of them. You move by inputting the number of the square you want to put your mark in. A key to the squares appears next to the board. The programs needs more than 1K.

```
10      DIM A(9)
20      LET Q = 0
30      FOR A = 1 TO 8
40      LET A(A) = 128
50      NEXT A
60      LET TR = 9
70      GOSUB 1000
80      LET TR2 = TR
90      LET TR = (TR + 1) - 8*(TR/8)
100     LET TR1 = TR
110     GOSUB 1000
120     IF TR = TR1 + 4 - 8*((TR1 + 3)/8) THEN GOTO  160
130     LET TR = TR1 + 4 - 8*((TR1 + 3)/8)
140     LET Q = 2
150     GOTO  1000
160     LET TR = TR1 + 2 - 8*((TR1 + 1)/8)
170     LET TR1 = TR
180     GOSUB 1000
190     IF TR = TR1 + 4 - 8*((TR1 + 3)/8) THEN GOTO 250
200     LET TR = TR1 + 4 - 8*((TR1 + 3)/8)
230     LET Q = 2
240     GOTO  1000
250     IF NOT 2*(TR2/2) = M2 THEN GOTO 290
260     LET TR = TR1 + 7 - 8*(TR1 + 2)/8)
270     LET Q = 2
280     GOTO 1000
290     LET TR = TR1 + 3 - 8*((TR1 + 2)/8)
300     LET TR1 = TR
310     GOSUB 1000
320     IF TR = TR1 + 4 - 8*((TR1 + 3)/8) THEN GOTO 360
330     LET TR = TR1 + 4 - 8*((TR1 + 3)/8)
340     LET  Q = 2
350     GOTO 1000
360     LET TR = TR1 + 6 - 8*((TR1 + 5)/8)
370     LET Q = 1
                              (NOTE: You can get this to run
                               in 1K by deleting all the blank
                               PRINT lines, and compressing
                               the PRINT statements.)
1000    CLS
1010    PRINT
1020    PRINT
1030    PRINT
1040    PRINT ,"MY MOVE * ";TR
1050    LET A(TR) = 180
1060    PRINT
1070    PRINT ,"1 * 2 * 3",CHR$(A(1));" * ";CHR$(A(2));" * ";
                                             CHR$(A(3))
1080    PRINT
1090    PRINT ,"8 * 9 * 4",CHR$(A(8));" * ";CHR$(A(9));" * ";
                                             CHR$(A(4))
1100    PRINT
1110    PRINT ,"7 * 6 * 5",CHR$(A(7));" * ";CHR$(A(6));" * ";
                                             CHR$(A(5))
1120    PRINT
1130    IF Q = 1 THEN PRINT ,"ITS A DRAW"
1140    IF Q = 2 THEN PRINT ," * * I WIN"
1150    IF Q> 0 THEN STOP
1160    PRINT ,"YOUR MOVE, HUMAN?"
1170    INPUT TR
1180    IF NOT A(TR) = 128 THEN GOTO 1170
1190    LET A(TR) = 189
1200    RETURN
```

SUBTERFUGE 144

"It", whatever it is, hides somewhere on one of 144 squares on a 12 X 12 grid. A wrong guess is rewarded by POKEing a "N" (for no) into the wrong location. This simple program is just crying out to be elaborated. Try and introduce some feedback for the player, perhaps related to the letter which is POKEd into the screen. SUBTERFUGE 144 needs 1K

```
10      LET H = 16396
20      LET A = RND(12)
30      LET B = RND(12)
40      PRINT
50      PRINT
60      FOR C = 1 TO 12
70      IF C = 1 THEN PRINT " * 123456789012 * "
80      PRINT ,,,,
90      NEXT C
100     PRINT ," * 123456789012 * "
110     FOR F = 1 TO 10
120     INPUT D
130     INPUT E
140     IF D = A AND E = B THEN GOTO 210
150     POKE D*33 + E + 1 + PEEK(H) + PEEK (H + 1)*256, 20
160     POKE 462 + PEEK(H) + PEEK( H + 1)*256, 156 + F
170     NEXT F
180     CLS
190     PRINT ,"SORRY, TIME IS UP"
200     GOTO 240
210     CLS
220     PRINT "YES, YOU FOUND IT"
230     PRINT "IN JUST * ";F;" * TRIES"
240     PRINT "IT WAS AT * ";A;" * ";B
250     PRINT ,"ANOTHER GAME?"
260     INPUT A$
270     CLS
280     IF CODE (A$) = 62 THEN RUN
```

BENJAMIN

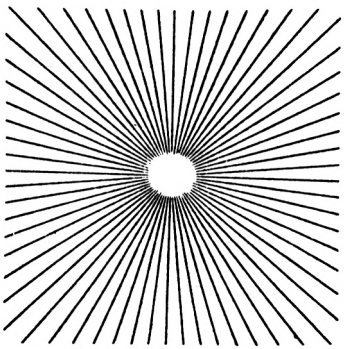
This no-nonsense program needs a strong NEWLINE finger. The computer rolls the dice, and uses them to build up pictures of poor, square-faced Benjamin. The winner is the player who gets Ben's face finished first.

```
10      RANDOMISE
20      LET A = 0
30      LET B = 0
40      LET C = 0
50      LET D = 0
60      FOR F = 1 TO 500
70      LET E = RND(2)
80      IF E = 1 THEN LET D = RND(6)
90      IF E = 2 THEN LET C = RND(6)
100     PRINT
110     PRINT ,"DICE * ";F;" * WAS FOR * ";E
120     PRINT
130     PRINT "PLAYER ONE"
140     PRINT "LAST ROLL * ";K,"WANTED * ";A + 1
150     IF C = B + 1 THEN LET B = B + 1
160     IF D = A + 1 THEN LET A = A + 1
170     LET G = A
180     GOSUB 300
190     PRINT
200     PRINT "PLAYER TWO"
210     PRINT "LAST ROLL * ";C;" * WANTED * ";B + 1
220     LET G = B
230     GOSUB 300
240     INPUT A$
250     FOR H = 1 TO 50
260     NEXT H
270     CLS
280     IF A$ = "" THEN NEXT F
290     STOP
300     IF G > 0 THEN PRINT ,CHR$(135);CHR$(131);CHR$(131);
                                CHR$(131);CHR$(134)
310     IF G > 1 THEN PRINT ,"shift Q  shift R  shift E";CHR$(130)
320     IF G > 2 THEN PRINT ,"shift Q space shift S space";CHR$(
                                CHR$(130)
330     IF G > 3 THEN PRINT ,"shift Q   -   ";CHR$(130)
340     IF G > 4 THEN PRINT ,CHR$(133);"shift W2;CHR$(132)
350     IF A < 6 AND B < 6 THEN RETURN
360     PRINT "THE GAME IS OVER"
370     PRINT "AND THE WINNER IS PLAYER * "; -1*(A = 6) -2*(B = 6
```

THE ENCHANTED FOREST

This game places you smack-dab in the middle, more or less, of a forest divided into an infinite number of triangular sectors. In one of them hides The Dragon. In others there are goblins, and fairies inhabit other sectors. You win the game by shooting the dragon. To shoot into, say sector 35, just input — 35.

This is the way it works. You start off in a random sector, and are given a choice of sectors into which you can move. If the dragon or a goblin is in the sector in which you land, you are rather dead. If you land on fairies, you will be transported at random to another sector. The fairies, goblins and the dragon do not move about during a game, so you can, if you don't get killed in the meantime, work out where they all are. You have just three arrows and you kill the dragon by shooting it from an adjoining sector. Start off you exploration of the enchanted forest by sticking to sectors 35 to 65, as you'll have the greatest chance of finding the dragon there.



```
10      DIM A(30)
20      DIM B(10)
30      LET G = 3
40      FOR Z = 0 TO 30
50      LET A(Z) = 0
60      NEXT Z
70      FOR Z = 0 TO 10
80      LET A = RND(30)
90      IF A(A) = 1 THEN GOTO 80
100     LET A(A) = 1
110     LET B(Z) = A + 35
120     NEXT Z
130     PRINT "FAIRIES HERE"
140     INPUT A$
150     LET X = 34 + 2*RND(16)
160     LET Y = 7
180     CLS
190     PRINT "YOU ARE NOW IN SECTOR * ";
200     LET A = -1
210     FOR Z = 0 TO 10
220     IF B(Z) = X THEN LET A = Z/5
230     NEXT Z
240     IF A = 0 THEN GOTO 130
250     IF A = 1 THEN PRINT "AND THE GOBLINS HAVE KILLED YOU"
260     IF A = 2 THEN PRINT " 3shiftQ YOU FOUND THE
                        DRAGON 3shiftQ "
270     IF A > 0 THEN STOP
280     PRINT "...YOU CAN MOVE TO * ";X - 1;" * ";X + 1;
                        " * ";X + Y
290     FOR Z = 0 TO 2
300     LET A(Z) = 0
310     NEXT Z
320     FOR Z = 0 TO 10
330     LET D  = B(Z) - X
340     IF ABS(D) = 1 OR D = Y  THEN LET A(Z/5) = 1
350     NEXT Z
360     LET D = ABS(D)
```

```
37Ø     IF D = 2 OR D = 6 OR D = 8 THEN LET A(2) = 1
38Ø     IF A(Ø) = 1 THEN PRINT "+++FAIRIES NEARBY+++"
39Ø     IF A(1) = 1 THEN PRINT "***GOBLINS NEARBY***"
4ØØ     IF A(2) = 1 THEN PRINT "*+* DRAGON NEARBY *+*"
41Ø     LET A = 2
415     PRINT "MOVE?"
42Ø     INPUT M
43Ø     IF M   Ø THEN GOTO 47Ø
44Ø     LET X = M
45Ø     LET Y = - Y
46Ø     GOTO 18Ø
47Ø     IF M = -B(1Ø) THEN GOTO 26Ø
48Ø     LET G = G - 1
49Ø     PRINT ,G;" * ARROWS LEFT"
5ØØ     IF G > Ø THEN GOTO 42Ø
```

SPIRALS

SPIRALS makes very, very effective use of the 1K on the old ROM ZX80, by POKEing an unusual playing board directly into PRINT statements. Your aim in this program is to get to the centre of the spiral in the shortest possible time. If you bump into a CHR$(128) you must back out. You cannot barge through. There is some very interesting coding which puts the score on the screen (see the routine starting at line 43Ø).

```
1Ø      PRINT " * * SPIRALS * "
2Ø      POKE 16421, 24
3Ø      PRINT
4Ø      PRINT  "..........."  (Note: These are full
5Ø      PRINT  ". + . 7spaces"       stops)
6Ø      PRINT  ". * . * ..... * ."
7Ø      PRINT  ". * . * . * * * . * . "
8Ø      PRINT  ". * . * . * . * . * . "
9Ø      PRINT  ". * . * . . . * . * . "
95      PRINT " . * . * * * * * . * . "
1ØØ     PRINT  ". * . . . . . . . * . "
11Ø     PRINT  ".   9spaces  ."
12Ø     PRINT  ". . . . . . . . . . "
13Ø     FOR P = 16460 TO 16623
14Ø     IF PEEK ( P ) = 27 THEN POKE P, 128
15Ø     NEXT P
16Ø     POKE 16429, 212
17Ø     POKE 16437, 212
```

RUN the above program, which will POKE CHR$(128)'s directly into PRINT statements. SAVE this, just in case something goes wrong later, then INPUT the following.

```
13Ø     PRINT
14Ø     PRINT "YOUR SCORE IS 9999"
15Ø     POKE 16414, Ø
16Ø     POKE 16415, Ø
17Ø     LET X = 26
18Ø     LET A = Ø
19Ø     IF A = 66 THEN GOTO 53Ø
2ØØ     INPUT N
21Ø     IF N = 5 THEN LET Y = X - 1
22Ø     IF N = 6 THEN LET Y = X + 12
23Ø     IF N = 7 THEN LET Y = X - 12
24Ø     IF N = 8 THEN LET Y = X + 1
25Ø     IF A AND Y - A THEN GOTO 43Ø     (Note: Enter these
26Ø     IF A THEN GOTO 35Ø                as listed)
27Ø     LET U = X
28Ø     LET V = Ø
29Ø     GOSUB 51Ø
3ØØ     IF PEEK (PEEK(16396) + PEEK(16397)*256 + Y) - 128
                THEN GOTO 39Ø
31Ø     LET A = X
32Ø     LET U = Y
33Ø     LET V = 147
335     GOSUB 51Ø
34Ø     GOTO 42Ø
35Ø     LET A = Ø
36Ø     LET U = X
37Ø     LET V = 128
38Ø     GOSUB 51Ø
39Ø     LET U = Y
4ØØ     LET V = 19
41Ø     GOSUB 51Ø
42Ø     LET X = Y
43Ø     LET S$ = STR$(9999 - PEEK(16414) - PEEK
                (16415)*256)
44Ø     FOR J = 1 TO 4
45Ø     LET U = 147 + J
46Ø     LET V = CODE S$
47Ø     GOSUB 51Ø
48Ø     LET S$ = TL$(S$)
49Ø     NEXT J
5ØØ     GOTO 19Ø
51Ø     POKE PEEK (16396) + PEEK(16397)*256 + U, V
52Ø     RETURN
53Ø     PRINT
54Ø     PRINT "GAME OVER"
```

Index

EXPLOSIVE GAMES FOR THE ZX81

RIP-ROARING GAMES FOR THE ZX80

(small page numbers)

* * * * * * * * * * * * * * * * * * * *

JOIN THE USERS' CLUB HERE

( ) Please register me as a member of the National ZX80 and ZX81 User's Club, and send me the next 12 issues of the monthly club magazine INTERFACE. I enclose £8.50 (UK), £11.50 (Europe), £14.50 (elsewhere).

( ) THE GATEWAY GUIDE TO THE ZX81 AND ZX80 — Mark Charlton — £5.95

( ) GETTING ACQUAINTED WITH YOUR ZX81 — Tim Hartnell — £4.95

( ) MASTERING MACHINE CODE ON YOUR ZX81 OR ZX80 — Tony Baker — £5.95

National ZX80 and ZX81 Users' Club, 44–46 Earls Court Road, London, W8 6EJ (mail order address only).

Please send me the indicated items. I enclose cheque for £_____, made out to NATIONAL ZX80 AND ZX81 USERS CLUB.

I have put my name, address and cheque card number on the back.

Name .....

Address .....

.....

(Send this page or a copy).

This book has listings for every game we thought you might want, including GALACTIC INTRUDERS, BREAKOUT, DRAUGHTS/CHECKERS, STAR TREK, DEATH MAZE, 4-IN-A-ROW and an 8K ADVENTURE-type program SMUGGLERS BOLD. As well, there are a host of new games, and adaptations of old favourites. Many of the programs will run in just 1K (including a simplified SPACE INVADERS-type program).

Some of the games are based on chance — the dreaded Sinclair random number generator — and others depend on skill, both yours and the computer. But we've tried to ensure that each and every program contains at least one programming technique which you'll be able to adapt for your own programs.

You can, if you like, just enter the programs as listed, and play them. However, you're likely to get much more enjoyment from working with them, altering them as you choose, deleting some sections, improving others, and so on, until the game carries your personal stamp. Many of the 1K games can be improved if you have extra memory. At the very least, the player prompts can be made more "user-friendly", and the rules explained more exactly.

AMSTRAD CPC

MÉMOIRE ÉCRITE
MEMORY ENGRAVED
MEMORIA ESCRITA

https://acpc.me/